

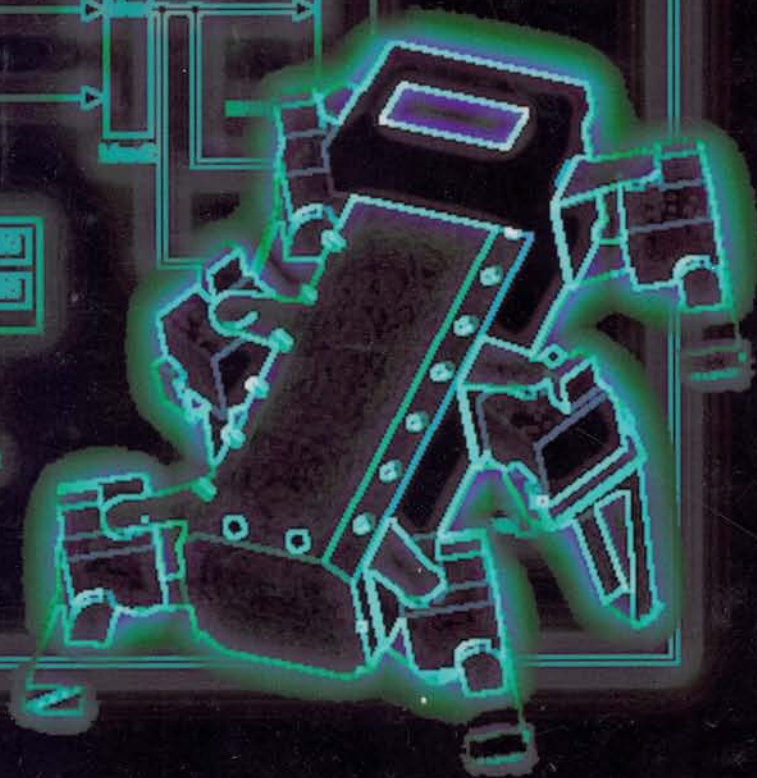


Universidade da Coruña

DEPARTAMENTO DE INGENIERÍA INDUSTRIAL II

Tesis Doctoral

ENTORNO EVOLUTIVO DE DISEÑO AUTOMÁTICO EN INGENIERÍA



Adolfo Lamas Rodríguez



UNIVERSIDADE DA CORUÑA

ESCUELA POLITÉCNICA SUPERIOR

Departamento de Ingeniería Industrial II

TESIS DOCTORAL

ENTORNO EVOLUTIVO DE DISEÑO AUTOMÁTICO EN INGENIERÍA

Autor: Adolfo Lamas Rodríguez
Director: Richard Duro Fernández
Fecha: Julio de 2004



UNIVERSIDADE DA CORUÑA


ESCUELA POLITÉCNICA SUPERIOR

Departamento de Ingeniería Industrial II

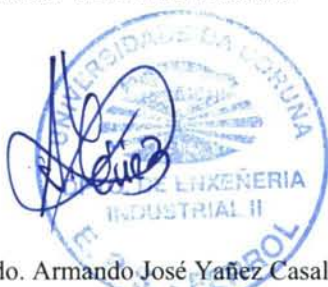
D. Richard J. Duro Fernández, Profesor Titular de Universidad del Departamento de Computación de la Universidade da Coruña,

CERTIFICA:

Que la memoria titulada “Entorno Evolutivo de Diseño Automático en Ingeniería” ha sido realizada por D. Adolfo Lamas Rodríguez bajo mi dirección en el Departamento de Ingeniería Industrial II de la Universidade da Coruña, y constituye la Tesis que presenta para optar al grado de Doctor.


Fdo. Richard J. Duro Fernández
Director de la Tesis Doctoral


Fdo. Armando José Yañez Casal
Director del Departamento de Ingeniería Industrial II



Agradecimientos

Durante estos años son muchas las personas e instituciones que han participado en este trabajo y a quienes quiero expresar mi gratitud por el apoyo y la confianza que me han prestado de forma desinteresada.

En primer lugar agradezco la acogida, el apoyo y los medios recibidos en el Departamento de Ingeniería Industrial II de la Escuela Politécnica Superior de la Universidade da Coruña sin los cuales no hubiera sido posible la realización de esta tesis.

Mi más sincero agradecimiento a mi director de Tesis, Richard Duro Fernández, por su valiosa contribución científica, su apoyo, confianza y permanente actitud positiva que me han servido como estímulo para el desarrollo de este trabajo. Sin su inestimable esfuerzo la financiación de esta tesis hubiera sido imposible.

No puedo olvidar a mis compañeros y amigos del laboratorio del Grupo de Sistemas Autónomos, (Juan, José Antonio y Fran) con los que he compartido incontables horas de trabajo. Gracias por tantos buenos momentos, por ayudarme, aguantarme y escucharme.

Quiero agradecer de igual modo las valiosas sugerencias y aportaciones de José Antonio Pérez Rodríguez y a la colaboración inestimable de Juan Carlos Becerra Permuy en el diseño de la portada. Al mismo tiempo, no puedo olvidar la ayuda prestada por D. Jesús Fernández Ibarz en el modelado del catamarán sumergible cuyo estudio pertenece a esta tesis.

Todo esto nunca hubiera sido posible sin el interés, comprensión y apoyo incondicional de mi familia, compañeros y amigos. Todos y cada uno de ellos han hecho más fáciles estos años de trabajo.

Les dedico este trabajo a mis padres por educarme, quererme y apoyarme en todo lo que he querido hacer en esta vida. Les agradezco que siempre hayan creído en mí y que en todo momento me hayan empujado a seguir adelante. Soy afortunado por contar siempre con su comprensión y ejemplo. Esta tesis es suya.

Quiero hacer mención especial a mi hermano por la amistad y los sueños que siempre hemos compartido y a mi ahijado Alejandro para que le sirva de estímulo en la vida que acaba de empezar.

Agradezco a María por su apoyo, paciencia, comprensión y afecto. Gracias por acompañarme en todos los momentos importantes de mi vida y en los quedan por llegar.

Asimismo dedico esta tesis a mis amigos con los que siempre he compartido mis alegrías, inquietudes, éxitos y fracasos. En todo momento los llevo conmigo.

Publicaciones

R. J. Duro, J. Santos, F. Bellas, A. Lamas (2000). "On Line Darwinist Cognitive Mechanism for an Artificial Organism". Proceeding Supplement Book SAB2000, pp 215-224, ISBN 0-9704673-0-3.

F. Bellas, A. Lamas, R.J. Duro (2001). "Evolutionary Cognition for Autonomously Obtaining Primal Behaviors". Advances in Signal Processing, Robotics and Communications, pp 398-404, ISBN 960-8052-42-4.

F.Bellas, A.Lamas, R.J.Duro (2001). "Adaptive Behavior Through a Darwinist Machine". Advances in Artificial Life, pp 86-90, ISBN 3-540-42567-5.

A. Lamas, R.J. Duro (2001). "Integrating Advanced Simulators for the Automatic Design of Controllers for Underwater Vehicles". Proceeding Supplement Book WSES/IEEE 2001: SIM, pp 404-408.

F.Bellas, A.Lamas, R.J.Duro (2001). "Multilevel Darwinist Brain and Autonomously Learning to Walk". Proceedings of the Internacional Conference on Computacional Intelligence, Robotics and Autonomous Systems CIRAS 2001, pp 392-397, ISSN 0219-6131.

A. Lamas, R.J. Duro (2001). "ADEUS: Integrating Advanced Simulators, Evolution and Cluster Computing for Autonomous Robot Design". Proceedings of the Internacional Conference on Computacional Intelligence, Robotics and Autonomous Systems CIRAS 2001, pp 398-403, ISSN 0219-6131.

A. Lamas, R.J. Duro, F.López Peña. "Automating the design process through evolution with ADEUS". Proceedings of ESDA 2002, pp 1-6.

A. Lamas, J. Fernández Ibarz, R.J. Duro. "Diseño Evolutivo de un Control Neuronal para Catamaranes Submarinos". Actas del II Congreso español sobre Metaheurísticas, Algoritmos Evolutivos y Bioinspirados. Universidad de Oviedo. Gijón 2003, pp 108-114, ISBN 84-607-65-26-1.

A. Lamas, R. J. Duro (2003). "A Tool for the Automatic Design of Electronic Control Systems and Circuits for Manufacturing Plants". Proceeding of Second IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IDAACS' 2003, pp 236- 240, ISBN 0-7803-8138-6.

A. Lamas, R.J. Duro (2003). "A Tool for the Automatic Design of Electronic Control Systems and Circuits for Manufacturing Plants". International Scientific of Workshop on Computing 2003, Vol.2, Issue 2, pp 76-81. ISSN 1727-6209.

J. Fernández Ibarz, A. Lamas, F.López Peña (2003). "Study of the Dynamic Stability of an Underwater Vehicle". Proceeding Supplement Book of Stability of Ships and Ocean Vehicles, 8th International Conference. STAB 2003, pp 129-140, ISBN 84-688-2742-8.

Índice

1	Introducción.....	1
2	El Proceso de Diseño.....	11
2.1	El Proceso de Diseño en Ingeniería	11
2.2	Metodología del Diseño en Ingeniería	12
2.2.1	Metodología clásica.	13
2.2.2	Corriente Anti-metodológica	15
2.2.3	Metodología Moderna.....	15
2.3	Etapas del proceso de diseño.....	16
2.3.1	Definición de un problema.....	17
2.3.2	Análisis del problema.....	18
2.3.3	Síntesis del diseño	19
2.3.4	Análisis del diseño	21
2.3.5	Especificación de una solución. El ciclo de diseño.....	22
2.4	Herramientas computacionales de diseño.	23
2.4.1	Modelado de sistemas	24
2.4.1.1	Abstracción	24
2.4.1.2	Proceso de modelado	24
2.4.1.3	Clasificación de modelos	24
2.4.2	Simulación	26
2.4.2.1	Simulación vs. soluciones analíticas	27
2.4.2.2	Tipos de simulación	29
2.4.2.3	Aplicaciones y limitaciones	30
2.4.3	Tipo de herramientas computacionales.....	31
2.4.3.1	Herramientas de dibujo y modelado de sólidos	31
2.4.3.2	Herramientas de análisis	31
2.4.3.2.1	Herramientas de análisis de propósito general.....	31
2.4.3.2.2	Herramientas de análisis de propósito especial.....	32
3	Computación Evolutiva. Algoritmos Evolutivos.	39
3.1	Características Algoritmos Evolutivos.....	40
3.2	Técnicas de evolución artificial	42
3.2.1	Algoritmos Genéticos (AG).....	43
3.2.1.1	Fundamentos biológicos	44
3.2.1.2	Fundamentos matemáticos	46
3.2.1.2.1	Teorema fundamental.....	46
3.2.1.2.2	Convergencia	47
3.2.1.2.3	Exploración y explotación.....	48
3.2.1.3	Algoritmo Genético Simple (AGS).....	48
3.2.1.4	Implementación de un Algoritmo Genético.....	49
3.2.1.5	Algoritmos Genéticos Avanzados.....	52
3.2.2	Estrategias Evolutivas	53
3.2.3	Programación Evolutiva.....	54
3.2.4	Programas de Evolución	54
3.2.5	Programación Genética	54
3.2.6	Algoritmos Miméticos	55
3.2.7	Algoritmos Macroevolutivos	55
4	Diseño Evolutivo por ordenador	59
4.1	Tipos de Diseño Evolutivo.....	61
4.1.1	Diseño Evolutivo para Optimización.....	61
4.1.1.1	Optimización Específica de Diseños.....	62
4.1.1.2	Optimización Genérica de Diseños.....	63

4.1.2	Diseño Evolutivo Creativo.....	64
4.1.2.1	Diseño Evolutivo Conceptual	65
4.1.2.2	Diseño Evolutivo Creativo.....	65
4.1.3	Arte Evolutivo.....	68
4.1.4	Formas de Vida Artificial Evolutiva.....	70
4.2	Diseño Evolutivo híbrido	72
4.2.1	Diseño Evolutivo Integral.....	72
4.2.2	Diseño Evolutivo Creativo basado en la Vida Artificial.....	72
4.2.3	Vida Artificial Evolutiva Estética.....	73
4.2.4	Diseño Evolutivo Estético.....	73
5	Entorno Evolutivo de Diseño Automático.....	77
5.1	Punto de partida: Etapas del proceso de diseño	78
5.2	Características deseadas de un entorno de diseño.....	82
5.3	Bloques constructivos básicos	84
5.3.1	Módulo de búsqueda de soluciones	87
5.3.2	Módulo de toma de decisiones. Simulación/Interacción.	89
5.3.3	Módulo de distribución computacional.....	91
5.4	Conclusiones sobre el entorno de diseño.	94
5.5	Experimentos y Aplicaciones.....	95
6	Robots submarinos.....	101
6.1	Vehículos submarinos.....	101
6.1.1	Clasificación de los sumergibles.....	102
6.1.2	Aplicaciones de los vehículos sumergibles.....	104
6.2	Diseño y modelado de vehículos submarinos.....	105
6.3	Control de vehículos submarinos.....	106
6.3.1	Simulación	110
6.3.2	Modelos hidrodinámicos.....	110
6.3.3	Simuladores.....	113
6.3.4	Sensores	114
6.4	Diseño de Controladores para Vehículos Submarinos.....	115
6.4.1	Hovering.....	116
6.4.1.1	Hovering sin brazo excavador.....	118
6.4.1.2	Hovering con brazo excavador	122
6.4.1.3	Conclusiones del diseño de controladores para hovering	129
6.4.2	Control de Maniobras para Catamaranes Submarinos.....	129
6.4.2.1	Catamarán sumergible modelado y control	130
6.4.2.2	Geometría.....	131
6.4.2.3	Superficies de control.....	132
6.4.2.4	Criterio de signos para la deflexión de las superficies de control.....	133
6.4.2.5	Experimentos	139
6.4.2.6	Resultados	143
6.4.2.7	Conclusiones del diseño de controladores para maniobra	148
6.4.3	Conclusiones del diseño de controladores	148
7	Plantas industriales	153
7.1	Marco teórico. Simulación de Procesos.....	153
7.1.1	Sistemas vs. Procesos.....	153
7.1.2	Sistemas discretos. Sistemas de Fabricación.	155
7.1.3	Diseño y Optimización de Sistemas y Procesos	157
7.2	Diseño de Morfologías y Controladores para Plantas Industriales	158
7.2.1	Optimización automática de morfologías de planta.....	160
7.2.2	Optimización automática de controladores secuenciales.....	164

7.2.2.1	Línea de Barnizado y Secado.....	164
7.2.2.2	Modelado	179
7.2.2.3	Resultados del estudio tradicional.....	184
7.2.2.4	Conclusiones del diseño tradicional.....	187
7.2.2.5	Estado del Arte en Hardware Evolutivo.....	188
7.2.2.6	Implementación de la metodología evolutiva de diseño automático.....	191
7.2.3	Conclusiones del diseño de controladores	203
8	Diseño Automático de Morfologías 3D.....	207
8.1	Introducción	207
8.2	Marco teórico	207
8.2.1	CAD y el proceso de diseño.....	208
8.2.2	Evolución histórica del diseño gráfico.....	208
8.2.2.1	CAD clásico	208
8.2.2.2	Integración vertical de aplicaciones de diseño.....	210
8.2.2.3	Diseño interactivo.	211
8.2.2.3.1	Interfases de usuario.....	211
8.2.2.3.2	Evolución de interfaces de usuarios.....	212
8.2.2.4	Diseño Interactivo Evolutivo	212
8.3	Computación evolutiva interactiva	215
8.4	Metodología para el Diseño Automático de Morfologías 3D	215
8.4.1	Objetivos	215
8.4.2	Implementación.....	216
8.4.3	Conclusiones del diseño morfológico	223
9	Conclusiones.....	227
10	Referencias.....	237

Introducción

1 Introducción

El proceso de diseño en ingeniería se ha ido depurando en el transcurso de los años hasta materializarse en una metodología estandarizada sobre la que se han definido múltiples aproximaciones. A pesar de las discrepancias existentes, todas ellas hacen hincapié en la definición e interrelación de sus etapas y destacan la importancia de la decisión del diseñador humano. En este sentido, el criterio de selección de los caminos de búsqueda y la evaluación de soluciones necesarios para la resolución de un determinado problema queda exclusivamente en manos del diseñador. Tanto es así, que las herramientas software existentes en el mercado relacionadas con el proceso de diseño se centran casi exclusivamente en él, tratando de ayudarlo en la visualización de modelos y en uno de sus cometidos principales: el análisis.

Evidentemente, el mantener al humano intrínsecamente ligado al proceso de diseño implica tener en cuenta sus limitaciones (tiempo, atención y capacidad) cuando se trata con estructuras complejas, no lineales o multidimensionales. Otra de las consecuencias es la aparición de estrategias del tipo “divide y vencerás”, que aunque hacen más manejable el problema, introducen multitud de restricciones ficticias que limitan o incluso imposibilitan la obtención de soluciones.

Por lo tanto, se hace necesario liberar al humano de su intervención en la toma de decisiones (búsqueda y análisis) y proporcionar un entorno de diseño automático que sólo necesite de su colaboración en aquellos casos donde intervengan elementos subjetivos y siempre desde un nuevo punto de vista: el usuario. Esta necesidad aumenta si se pretende resolver problemas de alta complejidad, con acoplamientos no lineales entre los parámetros que definen su calidad (productividad, rentabilidad, etc.) y altamente incorrelados con los correspondientes a sus subsistemas (controladores). Del mismo modo, cuando la aptitud de las soluciones de un problema incluye subjetividad o se requiere de la interacción de múltiples fuentes de conocimiento distintas, la automatización del proceso de diseño resulta muy ventajosa.

El trabajo de esta tesis se centra en la resolución de este problema, es decir, en el estudio de las bases de una metodología que permita que el humano deje su función de diseñador para hacerse cargo únicamente de la etapa de especificación del problema. Todo esto es posible mediante la síntesis de un mecanismo no humano que deje el proceso de exploración del espacio de posibles soluciones en manos de un sistema automático de diseño y que permita resolver el problema con garantías en función de la experiencia previa, la creatividad, las especificaciones y los recursos de que se disponen.

Esta preocupación por el proceso de diseño en ingeniería nace como respuesta a la evolución de los sistemas de fabricación, el incremento de la competencia, el aumento de la complejidad del producto y una mayor exigencia en su calidad. A pesar de este avance, el concepto básico de diseño en ingeniería ha cambiado muy poco y se define como un proceso que utiliza conocimiento, recursos y productos existentes para crear nuevos bienes y procesos a través de elementos funcionales y estéticos. Este tipo de diseño se divide en dos categorías: el diseño de sistemas y el diseño de productos.

En el diseño de sistemas se hace hincapié en el conjunto de entidades (procedimientos, materiales, información, espacio y energía) y recursos (equipos y personal) que

interaccionan entre sí con el objetivo común de constituir procesos fundamentales para la síntesis de un producto. El diseño de productos, por otro lado, es una actividad compleja que incluye análisis de mercado, producción y servicio y cuyo objetivo es producir un bien rentable que satisfaga las necesidades del consumidor y que sea seguro para éste y el medio ambiente.

En ambos casos es evidente que, si para dar solución a un problema se concentran los esfuerzos exclusivamente en el conocimiento técnico y científico aportado por soluciones adoptadas con anterioridad a problemas similares, se limita en gran medida la posibilidad de encontrar otras más efectivas todavía no contempladas. Esto es lo que ocurre en el proceso de diseño tradicional en donde existe un excesivo apego a soluciones consideradas como estándar, se asume como inútil toda inversión en el método de búsqueda o, si éste existe, se basa en modificaciones relativamente pequeñas a partir de una solución similar hasta que un límite de tiempo o calidad del diseño da por terminado el proceso.

Este antiguo enfoque, únicamente válido para problemas relativamente simples, se distancia del concepto básico de proceso de diseño mediante el cual se pretende definir una guía general para la especificación de una solución satisfactoria en términos funcionales y económicos, y que es aplicable en ingeniería tanto a problemas sin precedentes como a los que sólo consisten en la aplicación del conocimiento a un caso particular. La necesidad de reenfocar el proceso de diseño hizo necesario definir una serie de metodologías que estudiaran y analizaran el proceso de diseño y sus etapas para estimular constantemente la creatividad, evitar la intervención del azar en la toma de decisiones y permitir la integración de todos los factores funcionales, culturales y de producción que participan en él.

Todas las corrientes metodológicas que se definieron concuerdan en que el proceso de diseño no consiste en resolver un problema con una única solución sino en generar un abanico de posibilidades considerando requisitos a veces contradictorios y obtener a partir de ellos una solución definitiva que permita un margen de especulación amplio fundamentado en la creatividad. De este modo, el proceso de diseño se convierte en un medio efectivo para proporcionar buenos resultados aunque sea imposible señalar una combinación óptima que asegure el éxito del proyecto.

Queda claro que la confianza excesiva en soluciones tradicionales es un camino fácil al que generalmente se acude cuanto más especializado sea el problema a resolver, pero que en todo caso resta oportunidades a soluciones impensables que puedan ser más efectivas. No obstante, el proceso de diseño creativo no es tan directo, inmediato y controlable como el que consiste en buscar soluciones existentes o realizar ligeras modificaciones sobre las mismas. Tanto es así que la síntesis y evaluación constante de soluciones innovadoras requiere de un gran esfuerzo a la hora de resolver un problema determinado en el tiempo disponible. En general, la posibilidad de lograr este tipo de soluciones es directamente proporcional a la cantidad, calidad y variedad de los conocimientos puestos en práctica y depende en gran medida del método de búsqueda adoptado.

La necesidad de buscar una amplia variedad de soluciones obliga a sistematizar el proceso de combinación de las soluciones parciales, la solicitud de sugerencias, la evaluación de las mismas, etc. De este modo, las posibles alternativas de una solución deben ser evaluadas y recombinadas en la siguiente fase del proceso de diseño hasta obtener una solución completa que sea la mejor de las soluciones parciales. Es de

interés, por tanto, evitar la aparición de restricciones ficticias que no dan oportunidad a soluciones potencialmente válidas y que reducen la zona de exploración en torno a la solución presente.

Otra pieza clave en el proceso de diseño, además de la búsqueda sistemática de soluciones para fomentar la creatividad, es la integración de conocimiento en el análisis y evaluación de las mismas. Este concepto parte de la idea de que la creatividad no es un concepto exclusivo del diseño básico de un producto o sistema. Hace algún tiempo el proceso de diseño preliminar y el de producción poco tenían que ver el uno con el otro pero la aparición del modelado por ordenador dio lugar a un nuevo enfoque que reúne a ambos: la ingeniería concurrente.

Esta nueva idea choca frontalmente con el enfoque lineal que presenta el proceso de diseño tradicional en ingeniería donde éste evoluciona iterativamente a través de sus distintas etapas. El diseño en ingeniería concurrente, por el contrario, combina características del proceso tradicional con un enfoque de equipo no lineal que involucra a ingenieros de diseño y producción, técnicos y gerentes que trabajan de manera conjunta en la resolución de un problema.

El modelado, la simulación y la ingeniería concurrente permiten que cualquier integrante del equipo pueda tener acceso al diseño en curso y sea posible que todas las áreas de una empresa trabajen de manera simultánea a medida que se desarrolla el producto. Por ejemplo, los ingenieros de diseño pueden crear un modelo preliminar en 3D al inicio del proceso, un ingeniero mecánico puede utilizar un simulador para analizar sus propiedades térmicas y los ingenieros de diseño utilizar la información obtenida para realizar los cambios necesarios.

Una vez que el proceso de búsqueda de soluciones haya sintetizado un abanico de posibilidades son necesarios gran cantidad de análisis funcionales, estéticos, financieros y de mercado antes de que el proceso de diseño afronte una de sus fases más críticas: la toma de decisiones. En esta etapa es necesario determinar, a partir de la calidad de las soluciones parciales candidatas a un problema, un camino hacia la convergencia de las mismas. Este proceso se simplifica en gran medida si el criterio de búsqueda creativo y sistemático definido evita cuantificar numéricamente cada una de las soluciones que se obtienen y la toma de decisiones, una vez realizada su evaluación, consiste en asignar como preferentes a unas en relación a otras,

Teniendo en cuenta estos conceptos, en esta tesis se defiende la automatización del diseño de sistemas y productos mediante una metodología de carácter creativo basada en técnicas evolutivas que integre todas y cada una de las etapas que conforman el proceso de diseño: definición y análisis de un problema, síntesis y análisis del diseño y especificación de la solución. Para ello, y con objeto de proporcionar una línea metodológica clara y establecer la necesidad de la originalidad de las soluciones aportadas, se dedica un primer capítulo a definir el campo en el que se enmarca este trabajo, esto es, el diseño en ingeniería desde un punto de vista estructural o formal. Esto lleva a realizar un somero repaso a los orígenes y principales aportaciones a este campo de otros autores a lo largo de su historia así como a proporcionar una descripción del estado actual del mismo, especialmente en el ámbito que nos interesa, que es el de su aplicación práctica en la industria.

Con respecto a esta aplicabilidad, se realiza además un estudio sobre las actividades que constituyen el proceso de diseño en ingeniería y que hasta el momento, en general, han

sido realizadas de forma manual por ingenieros basándose fundamentalmente en su experiencia particular. La mayor parte de la investigación desarrollada en este campo ha consistido en la creación de herramientas computacionales que ayudan al ingeniero en la visualización o modelado de diseños particulares con el objeto de simplificar su tarea de valoración y asistirle en el proceso, por el momento dejado en sus manos, de indicar los caminos a seguir o las modificaciones a realizar sobre el diseño para el cumplimiento de sus requisitos y basadas siempre en su experiencia.

Partiendo de la identificación de estas actividades, en un capítulo posterior, se presentan de forma individual posibles soluciones para su automatización que permitirían su implantación sin la participación del ingeniero una vez especificados los requisitos del sistema y los recursos disponibles.

Se estudian después las diversas formas de integrar los elementos planteados con el objeto de conformar un sistema único, global y flexible de diseño automático que permita a su usuario adaptarlo de forma fácil al problema de diseño sobre el que esté trabajando. Como consecuencia de las etapas anteriores, surge una aproximación metodológica clara para la adaptación del sistema de diseño a distintos ámbitos y una especificación precisa de los elementos a tener en cuenta y como combinarlos.

Como solución a todo esto, se define una metodología evolutiva de diseño automático que combina la fidelidad de simuladores avanzados con la flexibilidad y versatilidad de las técnicas evolutivas y la potencia de la computación distribuida para proporcionar las bases de una herramienta que pueda obtener diseños complejos que trabajan en el mundo real en poco tiempo y a bajo coste.

Con el objeto de demostrar la aplicabilidad del entorno expuesto se han implementado procesos de diseño con distintas especificaciones en tres ámbitos diferenciados. Es importante resaltar que estos ámbitos se han escogido por existir sistemas reales y proyectos en los que se estaba trabajando que permitían comprobar la eficacia de los diseños obtenidos.

Por una parte se ha considerado el diseño de estructuras de control basadas en redes neuronales artificiales para el control de dos sistemas submarinos diferentes, uno basado en thrusters y el otro controlado mediante superficies. El objetivo principal es demostrar como la metodología evolutiva de diseño automático desarrollada es capaz de obtener soluciones para problemas dinámicos complejos donde interactúan diversos controladores no lineales en un medio complicado a nivel de sistema, y que utilizando estrategias del tipo divide y vencerás típicas del diseñador humano resultan muy difíciles de obtener dadas las no linealidades de las interacciones entre los distintos componentes.

En primer lugar, se plantea la síntesis de controladores basados en redes neuronales artificiales para la estabilización en posición (hovering) de un AUV (Autonomous Underwater Vehicle, Vehículo Sumergible Autónomo) al que es posible incorporar opcionalmente un brazo manipulador excavador. El control diseñado actúa sobre un conjunto de propulsores en función de un sistema de detección, basados en acelerómetros que incorpora el AUV para la compensación de cualquier fuerza ejercida sobre él, incluyendo la reacción del movimiento del brazo excavador y los cambios en su peso cuando excava. Se pretende de este modo que su velocidad sea siempre cero en cualquier dirección y su posición permanezca inalterable.

Como segundo ejemplo se diseñan automáticamente controladores neuronales de maniobra para un catamarán sumergible. Los controladores actúan sobre los timones de profundidad y dirección y sobre el sistema de propulsión en función de las consignas de maniobra indicadas y de los datos proporcionados por los sensores de posición, velocidad y aceleración, procurando evitar el desequilibrio que supone la aparición de fuerzas perturbadoras.

En ambos casos se requiere de un número elevado de simulaciones hidrodinámicas hasta llegar a sintetizar controladores de calidad debido a la complejidad del sistema en cuestión. Esto hace necesario la utilización de una gran cantidad de recursos computacionales y por esta razón la implementación del sistema sobre un entorno de computación distribuido en un cluster de PCs supone una ventaja debido a su escalabilidad.

Como segundo caso de ejemplo de aplicación se pretende demostrar la versatilidad del entorno de diseño desarrollado aplicando su metodología a la disposición de maquinaria y control de plantas industriales. En concreto se optimizan parámetros económicos de carácter global tales como la productividad cuya relación con los elementos de control a participar es extremadamente indirecta.

Con respecto a la optimización morfológica, se estudia la optimización de la disposición en planta de la maquinaria que forma parte de una línea de montaje constituida por dos generadores de subproductos complementarios y de cadencias distintas.

En relación al control, se desarrolla mediante técnicas de hardware evolutivo un controlador secuencial óptimo que obtenga para una línea de barnizado en una industria real la máxima productividad con el menor gasto energético posible. Los controladores candidatos están constituidos por circuitos lógicos secuenciales diseñados mediante puertas lógicas nand y circuitos flip-flop tipo D codificados en árboles binarios. El controlador óptimo debe modular la velocidad de los elementos de transporte de la línea de manufactura para evitar la existencia de esperas o atascos en la misma en función de la sensorización establecida en el modelo de la planta.

Finalmente, y con el objeto de llevar las posibilidades de la metodología planteada al límite del ámbito de la ingeniería donde linda con el arte o con los aspectos sociológicos y subjetivos de la aceptación del producto, se ha planteado el diseño automático e interactivo de morfologías tridimensionales. El objetivo fundamental de esta implementación consiste en la integración del entorno de diseño automático desarrollado con un sistema CAD (Diseño Asistido por Ordenador) y una interfaz interactiva para la síntesis de formas en 3D mediante la evaluación múltiple y subjetiva por parte de usuarios. De este modo se sientan las bases necesarias para la creación de un nuevo sistema computacional interactivo y evolutivo de “ayuda al diseño” basado en un clásico programa CAD que “ayuda al diseñador”, con la diferencia fundamental de que no es necesario especificar las modificaciones a realizar ni la calidad de las soluciones parciales obtenidas si no simplemente ordenar su aptitud según le parezca aceptable al usuario o a los expertos de cualquier disciplina que puedan intervenir en el diseño.

La incorporación de parámetros subjetivos en la evaluación múltiple de un producto mediante una interfaz hombre-máquina puede ser realizada por expertos en cualquiera de las fases del diseño o por usuarios finales. Este tipo de ingeniería concurrente incrementa en teoría la aceptación del producto en el mercado desde el punto de vista

estético y ergonómico al evaluar el producto en función de su morfología, de sus especificaciones funcionales y del cumplimiento de normativas técnicas. Así, teniendo en cuenta que el éxito de un producto en el mercado depende del equilibrio de estos tres factores, la aplicación del entorno de diseño que se plantea en esta tesis proporciona las bases para una herramienta sencilla, escalable y económica con la que sería posible diseñar productos que se adapten al máximo a los gustos y necesidades de un perfil determinado de usuarios al que van dirigidos.

Estos tres capítulos de aplicación de la metodología planteada siguen una estructura similar. En ellos se introduce inicialmente el campo en el que se realizan los experimentos, indicando otros trabajos relacionados y explicando el problema y su dificultad. A continuación se presenta y justifica la combinación de elementos que se eligen para la realización del proceso de diseño automático en ese ámbito en particular y seguidamente se pasa a realizar diversos diseños automáticos mostrando los resultados de los mismos. Finalmente, el último capítulo presenta una serie de conclusiones y posibles líneas de trabajo futuro.

Concretando, dentro de la corriente investigadora consistente en liberar al humano de su intervención en la toma de decisiones (búsqueda y análisis) del proceso de diseño clásico, el objetivo principal de esta tesis consiste desarrollar un entorno automático que integre todas y cada una de sus etapas (definición y análisis de un problema, síntesis y análisis del diseño y especificación de la solución) en donde la participación del humano consista únicamente en la especificación del problema y la evaluación subjetiva de soluciones. En este sentido, es necesario integrar al sistema de diseño técnicas y herramientas software que sustituyan al diseñador en sus tareas de búsqueda y análisis de las soluciones parciales generadas por el entorno.

Ya existen sistemas de diseño automático que cumplen este objetivo pero en todo caso consisten en desarrollos exclusivos a problemas particulares que integran técnicas de búsqueda y análisis ad hoc. Como consecuencia de esta limitación, el segundo objetivo de esta tesis consiste en definir un entorno de diseño automático homogéneo y modular que permita la integración de una o varias herramientas de análisis comerciales avanzadas para dotar al sistema de versatilidad. De este modo, el entorno sería capaz de adaptarse fácilmente al problema de diseño en estudio analizando las soluciones parciales generadas en función de gran variedad de criterios (funcionales, estéticos, de mercado,...). La integración en el sistema de diseño de herramientas de todo tipo, ya sean comerciales o no, permite dotar al entorno automático de las características propias de la ingeniería concurrente tradicional en donde ingenieros de diseño, producción, técnicos y gerentes trabajan de manera conjunta en la resolución de un problema. Esto supone una gran ventaja.

Obviamente todo esto deja de tener sentido si no se trata de optimizar una de las etapas más importantes del proceso de diseño: la etapa de búsqueda. Partiendo del hecho de que es técnicamente inviable que la exploración de soluciones parciales se base en el azar o en la búsqueda de todo el espacio (sobre todo cuando se trata con problemas de cierta dificultad) se planteó como siguiente objetivo implementar una metodología de exploración automática robusta y versátil que permitiese adaptarse con facilidad al problema abordado. Esta metodología debe ser capaz de buscar soluciones sistemáticamente teniendo en cuenta las especificaciones, los recursos de que se disponen y la experiencia previa estimulando constantemente la creatividad y evitando la aparición de restricciones ficticias que no dan oportunidad a soluciones potencialmente válidas. Al mismo tiempo la metodología de búsqueda debe sistematizar

de igual modo el proceso de combinación de las soluciones parciales, la solicitud de sugerencias y su evaluación relativa hasta obtener del abanico de posibles soluciones un camino hacia la convergencia de las mismas.

Otro de los objetivos importantes que se marcaron al inicio de esta tesis fue posibilitar la interacción del entorno de diseño con un conjunto de usuarios que representasen el perfil del consumidor final del producto o a un grupo de expertos en alguna de sus aspectos técnicos. De este modo, se integra en el entorno un nuevo evaluador basado enteramente en la opinión subjetiva de un conjunto de humanos en función de las características técnicas o estéticas de las soluciones.

Por muy buenas que sean las características de la técnica de búsqueda implementada en el entorno no quita que en el caso de abordar el proceso de diseño de productos y sistemas de relativa complejidad sea necesaria una gran potencia de cálculo debido a las múltiples evaluaciones de las soluciones parciales que genera el sistema. Por tanto, la posibilidad de distribuir computacionalmente el entorno de diseño y obtener un sistema escalable y no dependiente del hardware particular es otro de los objetivos a alcanzar.

Como último objetivo y no menos importante se pretende implementar el entorno de diseño en distintos sistemas reales pertenecientes a industrias o proyectos de investigación para comprobar la eficacia de los diseños obtenidos.

El Proceso de Diseño

2 El Proceso de Diseño.

2.1 El Proceso de Diseño en Ingeniería

Según la Accreditation Board for Engineering and Technology [ABET, 86] la ingeniería es “una profesión en la cual, el conocimiento obtenido a través de las ciencias naturales y matemáticas se aplica con juicio, para desarrollar formas de utilizar económicamente los materiales y fuerzas de la Naturaleza en beneficio de la humanidad”.

Por otra parte, el diseño es el proceso de aplicación de la ingeniería para la creación de dispositivos, estructuras y procesos mediante la conversión de los recursos disponibles. Diseñar requiere de experiencia, sentido común, imaginación e ingenio.

A partir de estos dos conceptos se define el proceso de diseño, base fundamental sobre la que se sustenta esta tesis, como las actividades y eventos que constituyen una guía general necesaria para la especificación de una solución satisfactoria en términos funcionales y económicos una vez que ha sido definido un problema. Su aplicación es de carácter general y, de hecho, es factible para resolver problemas sin precedentes donde se requiera trabajo de investigación y también para aquellos de escasa originalidad que sólo consista en la aplicación de conocimiento anterior existente a un caso particular.

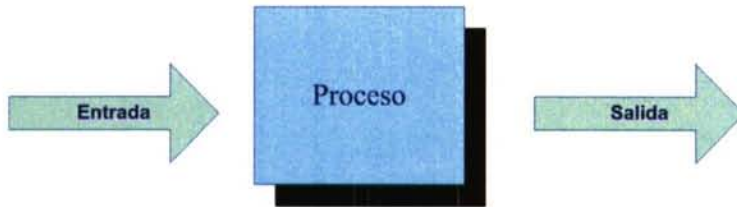


Figura 2.1: Proceso de diseño

El proceso de diseño consiste en una serie de etapas (definición y análisis de un problema, síntesis y análisis de soluciones y especificación de la solución adoptada) cuyas fronteras a veces parecen no estar bien definidas, creándose cierta confusión, aún cuando el objetivo de cada fase es bien concreto. Pueden aparecer nuevas soluciones mientras se aborda la fase de definición del problema, durante la fase de búsqueda podemos vernos tentados a una reformulación, etc.

Los diseñadores emplean un gran número de procedimientos de diseño, pero es inexacto asegurar que existe una combinación óptima. El seguir las reglas estrictas del proceso de diseño no asegura el éxito del proyecto y puede incluso inhibir al diseñador hasta el punto de restringir o limitar su imaginación. A pesar de esto, se cree que el proceso de diseño es un medio efectivo para proporcionar buenos resultados, organizados y útiles.

Con el objetivo de dar formato y estandarizar el problema de diseño se ha desarrollado toda una serie de corrientes metodológicas.

2.2 Metodología del Diseño en Ingeniería

En cualquier situación cotidiana se actúa mediante una determinada secuencia de acciones planificadas y aplicadas con mayor o menor lógica. En estos casos se puede decir que existe una metodología, es decir, un orden dentro de un conjunto de reglas que minimiza la intervención del azar en la toma de decisiones, y que siguiendo un camino más o menos reglado, nos hace llegar a un fin propuesto con antelación.

La aplicación general de la metodología sugiere aplicar sus preceptos al proceso de diseño donde surge la necesidad de determinados procedimientos para la toma de decisiones, algunos asimilados de situaciones anteriores, otros nuevos, pero en cualquier caso partiendo siempre de situaciones de información incompleta. De este modo se establecen estrategias que mediante un descubrimiento gradual reglado y procedimientos empíricos de valoración de soluciones respondan con la máxima eficacia posible a las necesidades del diseño.

Toda metodología trata del estudio y análisis del conjunto de procesos y métodos que se siguen en el diseño de un producto, su validación, uso y a las condiciones de elaboración de nuevos métodos. El método trata de un conjunto lógico de proposiciones o de fórmulas aplicadas al proceso en momentos determinados de éste, que permiten una acción y valoración frecuentemente simultáneas. Mientras que procedimiento es algo más material, método tiene un carácter mental y lógico.

La metodología aplicada al proceso de diseño supone la existencia de un control del mismo en cada etapa, un estímulo constante a la creatividad y una puesta en marcha de las necesidades actuales del trabajo en equipo. En ningún caso se puede asegurar que un determinado tipo de metodología suponga validez absoluta a los resultados. El mismo método puede ser correcto en unas situaciones e incorrecto en otras.

Las metodologías significan una toma de conciencia ante la magnitud de los sistemas de producción actuales y de los riesgos que un trabajo empírico e intuitivo significarían, dada la complejidad del mundo social y productivo a los que se dirige el diseño. Con los métodos se busca una definición del problema planteado desde un punto de vista lógico-científico y matemático, con cuyos recursos y técnicas se formaliza un sistema capaz de suministrar soluciones a dicho problema.

Existen diversas teorías metodológicas que han sido desarrolladas como solución al problema de diseño. Todas ellas parten del análisis exhaustivo de las condiciones del problema, continúan con los propósitos impuestos por el diseñador, estudian soluciones alternativas y finalmente posibilitan una propuesta definitiva. A continuación se hará una breve reseña histórica de la evolución de las corrientes metodológicas aplicadas al diseño.

Desde antiguo el ser humano ha buscado, de manera más o menos consciente, diversos modos de diseñar productos. Durante la Edad Media los artesanos unieron el arte y la técnica en un oficio y los cambios que se hacían eran lentos y simultáneos al trabajo. En el Renacimiento la ciencia e ingeniería propuso nuevos estudios para la mejora de los métodos existentes, pero nada evitó que desde el siglo XVII y, sobre todo, en la Revolución Industrial se produjese la separación entre arte y técnica. El avance de la técnica obligó a la aparición de nuevos métodos y a la normalización del proceso de diseño ante la necesidad de controlar los cada vez más complejos sistemas de

producción. En 1923 Theo van Doesburg [Van Doesburg, 23] afirmó: “Para poder construir un nuevo objeto, necesitamos un método, esto es, un sistema objetivo”.

El Bauhaus [Gropius, 61] fue la primera escuela que afrontó los problemas de diseño. Algunos de sus miembros consideraban los resultados de sus trabajos como producto de sus métodos. Los métodos de diseño posteriores fueron posibles gracias al uso de un conjunto de técnicas que se habían desarrollado con fines logísticos en la segunda guerra mundial y que poco a poco se fueron adaptando a la teoría de sistemas. A principios de la década de los años 60 comienza a generarse un creciente interés por los aspectos metodológicos del diseño que se genera inicialmente en los medios académicos ingleses y norteamericanos desarrollándose considerablemente en la década posterior y culminando con su institucionalización como materia universitaria.

Las principales corrientes metodológicas se pueden clasificar en tres grandes grupos: la metodología clásica, que engloba a su vez las vertientes científica y creativa, la antimetodológica y la moderna.

2.2.1 Metodología clásica.

En una conferencia sobre métodos de diseño en el Imperial College de Londres, Jones [Jones, 63] habló de las técnicas de diseño sistemático e inició el lanzamiento internacional del enfoque “metodológico”. Este nuevo enfoque pretendió paliar el relativo atraso del proceso de diseño en comparación con otras actividades técnico-científicas y se trató de descubrir su estructura y lógica interna.

El hilo conductor de la metodología clásica es la idea de la reducción de la complejidad. Los procedimientos racionalistas se adecuaban a esta necesidad y el pensamiento cartesiano se convirtió en el modelo a seguir [Descartes, 1637]. Este modelo dividía cada problema en tantas partes como fuese posible para resolverlo mejor, aumentando la complejidad del conocimiento paulatinamente y haciendo uso de revisiones.

Se evidenciaban una serie de tendencias en la propuesta de métodos de diseño: la corriente científica y la creativa.

- Corriente científica

Algunos métodos se basaron en ciertas disciplinas científicas como la Teoría de Sistemas en donde se analiza el problema como un todo y se estudian los problemas inherentes a sus partes, la Teoría de la Información, donde el problema es tratado como un mensaje, y la Teoría de Toma de Decisiones, fundamentada en la interpretación del resultado del conflicto mental que conlleva una elección.

La industria empezó a racionalizar cada vez más el proyecto, la construcción y la producción y los métodos creativos subjetivos y emocionales que procedían de la tradición del diseño artístico quedaron en segundo plano. El proceso de diseño se volvió más científico y el uso del ordenador se convirtió en una herramienta crucial.

En esta corriente se distinguieron Asimov, Archer, Alger, Hays, Alexander, Gugelot, Bürdeck y Jones. Los métodos de Archer y Asimov tuvieron un gran calado entre los profesionales. Por un lado Asimov [Asimov, 62] consideró dos grandes fases en el desarrollo de un método de diseño, la fase de planificación y morfología y la de diseño detallado: análisis, síntesis, evaluación y decisión, optimización, revisión y la

realización de prototipos. Por otro lado, el método de Archer [Archer, 63], fundamentado en el método científico, argumentaba que el proceso de diseño debía contener fundamentalmente las etapas analítica, creativa y de ejecución. A su vez estas etapas se subdividen en las siguientes fases:

- Definición del problema.
- Obtención de datos, preparación de especificaciones y retroalimentación.
- Análisis y síntesis de datos para la preparación de propuestas.
- Desarrollo de prototipos.
- Preparación de estudios y experimentos que validen el diseño.
- Preparación de documentos para la producción.

Alger y Hays [Alger, 64] estudiaron a fondo los procedimientos de valoración de las alternativas de diseño y propusieron las siguientes fases: definición del problema, especificación (obtención de requerimientos), proposición de alternativas, decisión, revisión y realización de prototipos.

Para Alexander [Alexander, 64] la clave se encontraba en el análisis riguroso del problema y en adaptar a éste la estructura del proceso de diseño y no a la inversa. Abogó por un racionalismo derivado de las ciencias exactas (análisis matemático) y empleó el método racionalista de análisis para subdividir el problema en subgrupos. La solución será la de aquellos problemas parciales relacionados progresiva y jerárquicamente. La implantación de estas exigencias es inabarcable sin la ayuda de ordenadores.

El método usado en la Escuela de Ulm fue desarrollado por Gugelot [Gugelot, 63] y fue el baluarte de esta metodología. En ella se sentaron los fundamentos de un acercamiento a la ciencia del proceso de diseño, aplicando algunos aspectos del pensamiento de Leibniz (1646-1716).

Jones [Jones, 63] no desarrolló propiamente un método pero destacó al definir su concepto de diseñador como “caja negra” o como “caja de cristal”. Algunas teorías, consideran que la parte más valiosa del proceso del diseño se realiza en el subconsciente del diseñador y que funcionando como “caja negra” es capaz de producir buenos resultados sin saber explicar cómo se obtuvieron. En otras, sin embargo, el procedimiento es más sistemático y racional (método analítico) y se denomina “caja de cristal”. En este proceso se fijan de antemano los objetivos, los criterios de evaluación y las estrategias, para posteriormente analizar el problema y buscar y evaluar lógicamente las soluciones.

- Corriente creativa

Tiene sus raíces en técnicas como la Tormenta de ideas [Osborn, 48], la Sinéctica [Gordon, 61], consistente en la analogía entre un problema sin resolver y otro similar con solución conocida, y el Pensamiento Lateral [Bono, 91]: para encontrar enfoques nuevos se ha de cambiar de percepción, desplazándose “lateralmente”. Hasta esta fecha los métodos empleados eran de carácter deductivo (del exterior al interior), pero los de esta nueva corriente son inductivos (del interior al exterior).

Maldonado [Maldonado, 77] y Dorfles [Dorfles, 77] hacen hincapié en la coordinación e integración de todos los factores que participan en el proceso: factores funcionales,

culturales y de producción (factores técnico-económicos, constructivos, de sistemas, productivos y de distribución). El método de trabajo no consiste en resolver un problema con una única solución sino en obtener gran número de ellas. Taboada y Nápoli [Taboada, 77] señalan la importancia de incorporar al proceso los propósitos que se desean lograr del producto. De las distintas combinaciones entre datos iniciales y propósitos surgirán diversas alternativas y, finalmente, se abordará una solución definitiva permitiendo un margen de especulación amplio (creatividad del diseñador). Esta creatividad no implica improvisación sin método, el propio método la estimula.

Las metodologías científicas tendían a separarse del arte y acercarse a las actividades científicas. Mañá [Mañá, 73] propone crear nexos entre la creatividad artística y la ingeniería. Bonsiepe [Bonsiepe, 85] estimaba que sólo dos métodos poseían valor instrumental para la actividad creativa: el método de la reducción de la complejidad de Alexander y el de la búsqueda de analogías o "Sinéctica" de Gordon.

2.2.2 Corriente Anti-metodológica

En torno a 1970 se pasó de la "racionalidad funcional" a la "irracionalidad más completa". Jones poco tiempo después de la publicación de su "Design Methods" [Jones, 70] argumentó que la lógica y la metodología son incompatibles con la naturaleza humana. En ese mismo año, Alexander [Alexander, 71] se da cuenta de que los métodos de diseño y su creación formal funcionalista destruyen la estructura mental que debe poseer el diseñador. Igualmente, en los años 80 se alzarán algunas voces señalando lo mismo, como es el caso de Tudela [Tudela, 80], para quien los métodos en diseño no aportan nada novedoso al producto.

2.2.3 Metodología Moderna

A mediados de los setenta aparecieron posturas contrarias a aceptar un método concreto como válido universalmente. La metodología de diseño no es algo absoluto y definitivo, es modificable. En opinión de Quarante [Quarante, 92] para cada problema planteado se necesitará un esquema metodológico previo y una planificación formando parte del estudio y variando en función del mismo. Esta misma idea la recoge Ricard [Ricard, 85] para quien en el diseño no hay un método, cada trabajo genera su método.

En los últimos años han aparecido nuevas orientaciones que proclaman un paso de las ciencias naturales a las filosóficas. Surgen preguntas, cómo, por quién, en que contexto, etc., se ha de emplear el producto. En este caso es evidente que el futuro de la metodología de diseño se encuentra en desarrollar sistemas que controlen y generen estrategias en colaboración con un buen número de disciplinas. El problema surge si se pretende realizar una metodología para la estética. Éste es un territorio poco desarrollado para la metodología.

A pesar de la variedad existente de corrientes metodológicas que pretenden establecer una secuencia de acciones a la hora de implementar el proceso de diseño, se convino en establecer las etapas que se describen en el siguiente apartado.

Las etapas fundamentales del proceso de diseño se pueden consensuar en las siguientes:

- Definición y análisis del Problema:
 - Reconocimiento de necesidades
 - Formulación del problema
 - Especificaciones criterios y restricciones
- Síntesis del diseño
 - Búsqueda de soluciones: recopilación de las soluciones alternativas mediante técnicas de indagación, invención, investigación, etc.
 - Selección de las alternativas más convenientes
 - Decisión: evaluación, comparación y selección de las alternativas existentes con el propósito de obtener la solución óptima.
- Análisis del diseño
 - Simulación mediante modelos matemáticos/computacionales
 - Prueba de modelos físicos
 - Optimización del diseño
- Especificación de la solución
 - Dibujos (esquemas) técnicos
 - Informes
 - Archivos computacionales (imágenes, textos, gráficos)
 - Datos de prueba
 - Maquetas
 - Simulaciones

2.3.1 Definición de un problema

El origen de un problema viene dado por la percepción de una necesidad y la búsqueda de un dispositivo físico, estructura o proceso que puedan satisfacerla. Una vez que se ha identificado la necesidad de un diseño, la definición de un problema requiere de una comprensión del mismo, una recopilación y análisis de los datos obtenidos mediante la comprobación de hipótesis y el establecimiento de relaciones causa-efecto y, por último, una formulación del problema en función del alcance de la solución que se está buscando con el objetivo de sintetizar de la mejor forma posible todo el proceso.

La formulación consiste en expresar de modo general la particularidad del problema a resolver prestando atención únicamente a la identificación del estado de entrada "A" y de salida "B". Pasar por alto la necesidad de formular un problema dotándolo de generalidad supone que ciertas soluciones meritorias sean excluidas. Asimismo, aunque en un caso real de complejidad sea necesario dividir el problema inicial en diferentes subsistemas, del mismo modo que el grupo humano que interviene en la búsqueda de la solución en equipos, esto puede limitar la efectividad de la solución obtenida.

La formulación de un problema puede realizarse a través de gran variedad de métodos. Entre ellos el método de la "caja negra" constituye una formulación esquemática de gran simplicidad y utilidad en el procesamiento de información y la resolución de problemas. En esta etapa de formulación del problema todo lo que suceda en el interior de dicha caja es desconocido y constituye los detalles que se tratarán en etapas posteriores del proceso de diseño.

El grado de amplitud en la formulación de un problema es de suma importancia y regirá en gran medida el desarrollo del proceso de diseño. Este valor aumenta sustancialmente la probabilidad de obtener soluciones de notable calidad y depende directamente de las limitaciones temporales y económicas asociadas al problema a resolver. La delimitación del problema y el alcance de la solución que se está buscando son cruciales para indicar lo que se quiere hacer y a dónde no se quiere llegar. Para ello es necesario conocer exhaustivamente y desde la mayor amplitud de perspectiva posible tanto el problema como la viabilidad de su resolución. Ésta es la parte más complicada en el proceso de diseño y una equivocación en este punto repercutirá en gran medida en la calidad del resultado final.

Además de la definición del propósito general, resulta de vital importancia para alcanzar el objetivo propuesto el proceso de traducción de unos requisitos vagamente especificados en un conjunto de especificaciones concretas. Las consideraciones económicas relativas al costo de desarrollo, realización y utilización de una solución, es decir, su viabilidad técnica y económica, son claves en el proceso de diseño. Un proyecto de ingeniería involucra irremediablemente restricciones con las cuales se debe trabajar, siendo una de las más importantes el costo. Las restricciones consisten, por tanto, en limitar inicialmente determinadas características de la solución, lo que supone reducir las posibles alternativas a seguir para resolver el problema. Aceptar automáticamente todas las restricciones como óptimas puede suponer una limitación considerable del desempeño de la solución obtenida. Una restricción ficticia consiste en la exclusión injustificada, indeseable y auto impuesta de una o varias posibilidades perfectamente legítimas. La eliminación de una restricción ficticia suele hacer que el problema admita soluciones más ventajosas.

En toda resolución de un problema existen preferencias a la hora de lograr la transformación deseada. Cada una de las normas que intervienen en la selección de soluciones se llama criterio. Por lo general la resolución de un problema de ingeniería presenta objetivos conflictivos y especificar el mejor balance entre los criterios en conflicto no es tarea fácil.

Es importante hacer hincapié en que de lo que se trata es de buscar metódicamente la solución a un problema mediante el proceso de diseño, evitando pretender encontrar la solución de forma inmediata simplemente optimizando una solución anterior ya existente. Existen múltiples caminos para lograr el propósito especificado, muchos de los cuales son desconocidos inicialmente para el ingeniero. Cabe resaltar la importancia de la exploración exhaustiva de todas las alternativas posibles en la búsqueda de una solución es muy importante y no debe sustituirse únicamente por la experiencia o conocimiento anterior. La imaginación y capacidad creativa debe ser una fuente de soluciones a tener en cuenta.

2.3.2 Análisis del problema

Así como para formular un problema se identificaban los estados inicial A y final B, durante el análisis del mismo se identifica, además, la transformación de un estado a otro. El análisis predice matemáticamente el rendimiento de un sistema mediante la determinación de las características cualitativas y cuantitativas de dichos estados. El carácter dinámico de estas características de los estados A y B da lugar que suelen denominarse variables de entrada y variables de salida, respectivamente.

La resolución de un problema proviene del hecho de intentar encontrar la transformación a partir de la cual desde un estado inicial de cosas o “estado A” es posible alcanzar otro estado final “estado B”. En la inmensa mayoría de los casos, existen gran cantidad de soluciones posibles para un mismo problema, muchas más de las que el tiempo dedicado al proyecto nos pueda permitir analizar. Todas y cada una de estas soluciones son un medio para lograr la transformación entre estados deseada.

2.3.3 Síntesis del diseño

Un problema de diseño responde a la pregunta inversa: “dado un cierto comportamiento buscado, cuál es el dispositivo o sistema que permite alcanzarlo”. Esto corresponde a la síntesis de un sistema. La solución de este problema inverso es un proceso iterativo, que involucra la solución de una serie de problemas directos mediante análisis.

Concentrarse inicialmente en intentar detallar una de las posibles soluciones a un problema dificulta en gran medida la capacidad de búsqueda de otras soluciones posiblemente de mayor calidad que la primera. En esta fase del proceso de diseño se conviene prestar atención únicamente a conceptos de solución, es decir, a las características generales de cualquier solución particular. Los detalles se tendrán en cuenta más adelante a la hora de tomar decisiones. Para afrontar el reto de dar solución a un problema es necesario tener en cuenta que además de existir un conocimiento técnico y científico previo no se debe desestimar la creatividad e inventiva propia. Lo importante es tener en cuenta que tanto el conocimiento de soluciones anteriores de problemas similares como la propia capacidad de inventiva para crear soluciones originales son fuentes de gran ayuda.

Aunque es imposible determinar todas las soluciones posibles de un problema, resulta conveniente apostar por la creatividad como estímulo para encontrar soluciones mejores que las obtenidas tradicionalmente. La confianza excesiva en soluciones existentes es un camino fácil, pero hay que tener en cuenta que por muy buena que se crea que es una solución concreta se restan oportunidades a otras originales que por sus características puedan ser de mayor calidad que las primeras.

Esto se debe a que la posibilidad de lograr soluciones efectivas es directamente proporcional a la cantidad, calidad y variedad de los conocimientos puestos en práctica. En general, cuanto más especializado sea el conocimiento, mayor será el número de soluciones tradicionales con que se estará familiarizado y más fácil será confiar excesivamente en ellas.

La inventiva es la capacidad de idear soluciones valiosas y depende de la actitud, aptitud, el esfuerzo y el método de búsqueda. Salvo efímeras ideas felices, la producción constante y obligada de ideas y, sobre todo, la comprobación de la calidad de las mismas, requiere de un gran esfuerzo para resolver un problema determinado en el tiempo disponible. Aún así, es de suponer que por muy innovador que se considere el problema que tenemos entre manos, siempre existan experiencias anteriores en el mismo campo que pueda servirnos como punto de partida. El proceso de innovación no es por supuesto tan directo, inmediato y controlable como el de buscar soluciones existentes de problemas similares.

Es importante hacer hincapié en el método de búsqueda de soluciones. Normalmente las soluciones alternativas se asemejan en gran medida al tipo de solución actual existente. Esto es debido a que en la práctica el proceso de búsqueda se inicia con modificaciones

relativamente pequeñas de la solución tradicional avanzando progresivamente hacia mejores soluciones, hasta que un límite de tiempo o perfección diese por terminada la búsqueda. Esta inflexibilidad y limitación se convierte en un obstáculo al pensamiento creativo. El excesivo apego a soluciones y procedimientos tradicionales y la tendencia generalizada a suponer inútil grandes inversiones en el método de búsqueda, da lugar a soluciones similares a la existente. El límite de la zona de búsqueda que en teoría se define por la existencia de restricciones que rechazan soluciones no válidas, se reduce injustificada e inadvertidamente debido al conocimiento limitado y a las restricciones ficticias que no dan oportunidad a soluciones potencialmente válidas. La eliminación de las restricciones ficticias, la depuración de las restricciones reales y la ampliación de los conocimientos correspondientes al problema, mantienen inalterada la zona de búsqueda inicial. Además de evitar límites ficticios en la zona de exploración es fundamental que la exploración sea lo más extensa posible y no se reduzca únicamente a la región más próxima a la solución presente. Un método para minimizar la probabilidad de no tener en consideración un gran conjunto de prometedoras posibilidades es crear muchas posibilidades para cada una de las variables de solución trabajando desde lo general a lo específico. Esta necesidad de buscar una amplia variedad de soluciones básicamente diferentes obliga a sistematizar el proceso de combinación de las soluciones parciales, solicitud de sugerencias, examen de conocimientos técnico-científicos, etc. Las posibles alternativas de una solución serán evaluadas y recombinadas en la siguiente fase del proceso de diseño hasta obtener una solución completa que sea la mejor combinación de soluciones parciales.

Cualquier método de organización de carácter abierto tal como son los de árboles de alternativas son medios eficaces para sistematizar el proceso de búsqueda de una amplia variedad de soluciones básicamente diferentes. Existen determinados métodos predominantemente aleatorios que proporcionan variabilidad a las soluciones. Un ejemplo notable de este tipo de técnicas es la tormenta de ideas cuyo propósito es crear una atmósfera que incite fácilmente a la afluencia de las mismas, sin importar lo más mínimo lo absurdas que éstas puedan parecer inicialmente. La utilidad de este método es clave pues trata de evitar la tendencia generalizada de aceptar exclusivamente como soluciones parciales válidas aquellas que sean simples modificaciones mínimas de una solución de referencia. De este modo, se incrementa notablemente la probabilidad de obtener algún provecho que alguna de estas soluciones parciales. Otro tipo de método de búsqueda de soluciones parciales son las analogías que toman como referencia soluciones de problemas análogos.

2.3.4 Análisis del diseño

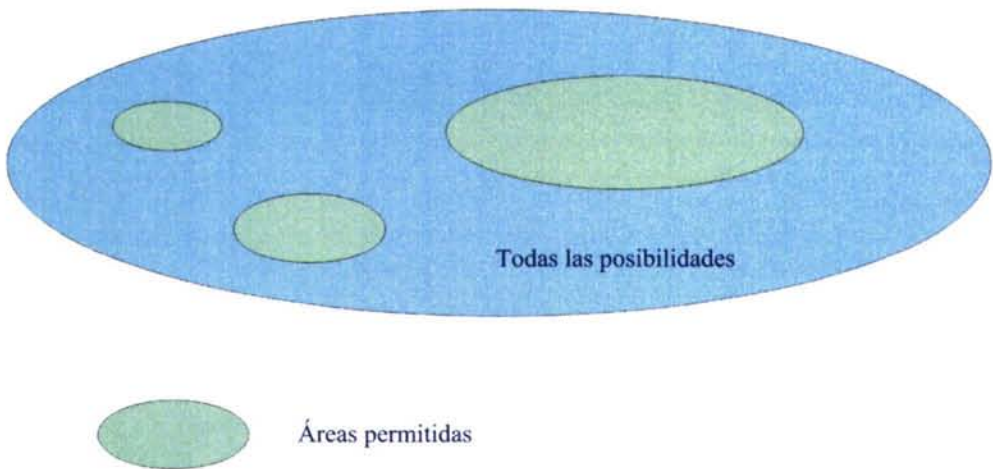


Figura 2.3: Análisis del proceso de diseño

Una vez que ha sido realizada la fase de búsqueda, en la fase de decisión del proceso de diseño se realiza la evaluación iterativa de las soluciones alternativas existentes otorgando mayor protagonismo a aquellas de mayor calidad que encaminarán el problema hacia la convergencia. Del mismo modo que la fase de búsqueda proporciona variedad a las posibles soluciones, es necesario que un método de decisión facilite la búsqueda de la solución considerada como óptima.

La decisión de detener el proceso para iniciar la evaluación de las soluciones parciales acumuladas no se toma cuando se haya encontrado la solución óptima por la dificultad del tiempo necesario para ello y sobre todo por la dificultad de saber si existe una única solución perfecta.

El proceso de toma de decisiones en ingeniería varía considerablemente según el tipo de problema, las técnicas que se empleen, la complejidad y competitividad de las alternativas, la importancia relativa de la decisión y otras circunstancias. Es posible desarrollar, perfeccionar y analizar varias ideas y cada una puede ofrecer ventajas sobre las otras. Resulta un grave inconveniente descartar ideas que, aunque inicialmente puedan no parecer ventajosas, podrían dar lugar a una solución excelente. En esta etapa hay que considerar toda solución como candidato a una evaluación posterior. La decisión acerca de cual diseño será el óptimo para una necesidad específica debe determinarse mediante experiencia técnica e información real. Siempre existe el riesgo de error en cualquier decisión, pero un diseño bien elaborado estudia el problema a tal profundidad que minimiza la posibilidad de pasar por alto una consideración importante, como ocurriría en una solución improvisada.

Esta fase de evaluación del desempeño o calidad de las soluciones alternativas se realiza mediante la experimentación con modelos y en función de los criterios seleccionados tras la determinación de su importancia relativa. El proceso debería detenerse cuando, en función de un cierto criterio y teniendo en cuenta siempre el tiempo disponible para ello, se haya alcanzado un punto en donde la calidad deje de tener un carácter creciente. El empleo de un criterio particular producirá diferencias de opinión respecto a si es

mejor partir de un gran número de alternativas iniciales en detrimento del tiempo de evaluación, a riesgo de no poder seleccionar la mejor, o bien asegurarse de haber elegido la mejor solución de una muestra más pequeña con una evaluación exhaustiva. Por lo general, el criterio predominante es la razón beneficio costo pues da idea de la utilidad de la inversión realizada. Sin embargo existen muchos otros como, por ejemplo, la fiabilidad, operabilidad (operación y mantenimiento) y disponibilidad del sistema.

La eficacia de las posibles soluciones de un problema no tiene porque estar directamente relacionada con su complejidad. Dentro de las posibles soluciones alternativas a un problema las más simples presentan indiscutibles ventajas económicas tanto de aplicabilidad como de mantenimiento. El esfuerzo por alcanzar la solución con las características más sencillas se verá en el futuro altamente recompensado. A veces es tipo de soluciones sorprenden o incluso decepcionan por su excesiva simplicidad y necesitan de un alto grado de inventiva y creatividad.

2.3.5 Especificación de una solución. El ciclo de diseño

La especificación de la solución comprende sus características físicas y funcionales que pueden ser representadas en innumerables formatos tales como planos, informes técnicos e incluso modelos tridimensionales físicos o virtuales. Esta etapa no representa el punto y final del proceso de diseño si no que éste ha de ser considerado como un ciclo que consta de los siguientes pasos:

- Aceptación y aplicación del diseño

Que la implantación de las soluciones ideadas sea estricta es algo con lo que no podemos contar sobre todo desde el punto de vista humano. Merece la pena prestar especial atención a la aceptación y aplicación plena de las propuestas adoptadas.

- Evaluación del desempeño

La inspección regular del desempeño de la solución resulta de gran ayuda a la hora de optimizar el proceso de diseño ya que el grado de calidad de la misma no permanece inalterable con el tiempo.

- Reactivación del proceso de diseño

La aparición en escena de nuevas técnicas, métodos y especificaciones, hace necesario la evaluación periódica de la solución y la valoración de una posible reactivación del proceso de diseño para el desarrollo de un nuevo concepto.

Una vez analizado el proceso de diseño y sus etapas se verá a continuación las herramientas computacionales existentes que ayudan a su implementación.

2.4 Herramientas computacionales de diseño.

Actualmente el ordenador se usa extensamente para dar soporte al proceso de diseño en sus diferentes etapas. Existe una gran cantidad de herramientas que asisten a este proceso, cada una con sus capacidades y limitaciones. A medida que se incrementa la duración del proceso del diseño, el costo necesario para ponerlo en práctica también lo hace. Por lo tanto, la utilidad de este tipo de software en su cometido de automatizar y acelerar el proceso queda fuera de toda duda. Además, debido al descenso incesante en el coste de los ordenadores, éstos se han convertido en una herramienta esencial para el diseñador.

La mayoría de los departamentos de diseño utilizan paquetes de software de diseño asistido por ordenador (CAD) para dibujar, manipular y almacenar sus diseños. Estas herramientas de software son cada vez más avanzadas, con capacidad para realizar renders fotorrealísticos, animaciones, o incluso generar mundos estereoscópicos de realidad virtual. Del mismo modo, las herramientas de análisis que simulan el funcionamiento de diseños también son cada vez más comunes y gran cantidad de procesos de diseños confían en este tipo de software para experimentar con los diseños antes de la construcción de prototipos.

Mientras que el CAD y otras herramientas de análisis computarizado son hoy frecuentes, el uso del ordenador para automatizar realmente el proceso de diseño (por ejemplo, mejorando diseños existentes) es actualmente menos común. Estas herramientas de software toman el control del proceso de diseño, permitiendo que los diseñadores mejoren el funcionamiento de sus diseños automáticamente, auxiliados por el software de análisis. La aplicación de este nuevo tipo de herramientas posibilitaría la generación de soluciones alternativas a un problema de diseño explorando numerosas soluciones creativas que superen las limitaciones del conocimiento convencional. Asimismo sería posible hacer uso del conocimiento previo de otros diseñadores en problemas semejantes al planteado, o incluso, sugerir conceptos de diseño completamente nuevos. La implantación de este tipo de herramientas es escasa y carece de versatilidad pues sólo existen soluciones específicas para cada tipo de problema en dominios muy limitados.

Es precisamente esta carencia de herramientas computacionales aplicadas al proceso de diseño el punto de partida de la concepción de la metodología evolutiva de diseño automático desarrollada en esta tesis. La búsqueda de un entorno computacional modular que aprovechara al máximo las herramientas existentes actualmente en el mercado para implementar ciertas fases del proceso e incluyese la posibilidad de incorporar al mismo tiempo módulos programados exclusivamente para el problema en cuestión fue considerada como premisa fundamental. La versatilidad del entorno fue otro de los pilares fundamentales para la síntesis del entorno, pues se pretendía desarrollar a partir de él todo tipo de sistemas, desde controladores hasta diseños en 3D.

2.4.1 Modelado de sistemas

La actividad principal de la Ingeniería es la resolución de problemas prácticos pero la experimentación con un sistema real puede, en ocasiones, no ser aceptable económicamente. El modelado supone un marco de conocimiento que facilita y economiza el análisis del sistema teniendo en cuenta el tiempo y dinero requeridos para su construcción y experimentación. Los productos con alta probabilidad de riesgo o incertidumbre, es decir, aquellos con alto costo de fallo o pertenecientes a nuevas tecnologías, se benefician en gran medida del uso de modelos.

Es necesario crear de modelos lógicos o matemáticos que permitan de manera simple el análisis predictivo del comportamiento del sistema real gobernado por las relaciones existentes entre sus variables.

2.4.1.1 Abstracción

Tal como se explica en [Rosenbluth, 45], ninguna parte sustancial de nuestro entorno es lo suficientemente simple como para que pueda ser comprendida y controlada sin abstracción. La abstracción consiste en reemplazar el sistema bajo consideración por un modelo de estructura similar pero más simple. Los modelos constituyen un elemento central del proceso científico.

Un modelo es, por tanto, una abstracción de la realidad que representa de forma simplificada un sistema real, de tal manera que captura la esencia funcional del sistema para poder predecir su comportamiento y control. Un modelo ha de tener el detalle suficiente como para que pueda utilizarse en la investigación y la experimentación en lugar del sistema real, con menos riesgo, tiempo y coste.

De acuerdo con Minsky [Minsky, 85], un objeto matemático es un modelo válido de un sistema para un observador, si es capaz de proporcionar respuestas válidas a las preguntas que el observador formula sobre el sistema.

2.4.1.2 Proceso de modelado

El modelado requiere de una metodología que comienza con la definición clara y concisa del problema delimitando sus fronteras. La elección del subconjunto óptimo de variables e interrelaciones que gobiernan el comportamiento del sistema es una cualidad que definirán la intuición y experiencia del modelador. Si este subconjunto predice con seguridad el comportamiento del sistema nos encontramos frente a un sistema determinista en contraposición a los sistemas estocásticos. Así se irá construyendo de manera interactiva, acumulando capas que pasan de un modelo básico a un modelo cada vez más complejo.

2.4.1.3 Clasificación de modelos

Los modelos pueden ser clasificados a lo largo de dos dimensiones. La primera dimensión es el grado en el cual un modelo es físico o analítico y la segunda el grado en el cual un modelo es global o parcial.

Los modelos físicos son artefactos tangibles para experimentación usados para validar la funcionalidad de un producto. En estos casos, un modelo físico puede servir como herramienta para detectar fenómenos perjudiciales que pueden maximizarse en el producto final. Los modelos analíticos representan productos no tangibles y generalmente se generan matemáticamente. Ejemplos de modelos analíticos son las simulaciones por ordenador, los sistemas de ecuaciones codificadas mediante una hoja de cálculo y los modelos tridimensionales realizados por ordenador. Esta aproximación matemática del producto está definida mediante parámetros. Una variación de uno de los mismos representa una posible alternativa de diseño.

En la mayoría de los casos, el modelo analítico es más flexible que un modelo físico, pero nunca puede revelar fenómenos que no son parte fundamental del mismo. Por esta razón, un modelo analítico frecuentemente precede a un modelo físico y es usado para estrechar el rango de parámetros factibles. El modelo físico se usa para poner a punto o confirmar el diseño.

Un modelo global implementa la mayoría, sino todos los atributos de un producto, corresponde al uso frecuente de la palabra modelo y es una versión a escala totalmente operacional del tamaño original. Un ejemplo de un modelo global puede ser aquel que se entrega al consumidor con el fin de que este identifique cualquier defecto de diseño antes de comenzar la producción. Un modelo parcial, en contraste, se centra en la implementación de uno o unos pocos atributos del producto. Algo común es usar varios modelos parciales juntos para analizar conjuntamente su apariencia física y comportamiento. Algunas veces los modelos contienen una combinación de elementos físicos y analíticos.

Al mismo tiempo, los modelos se pueden clasificar en los siguientes tipos:

- Determinísticos y estocásticos según el nivel de incertidumbre del sistema.
- Estáticos y dinámicos en función de la dependencia del sistema respecto al tiempo.
- Continuos y discretos dependiendo si un estado puede cambiar continuamente o sólo en determinados instantes.

El concepto de modelado fundamenta la relación existente entre el modelo y el sistema. Esta relación se instrumenta en el ordenador mediante la simulación. Los modelos de simulación son modelos analíticos que requieren de especial consideración.

La mayoría de los modelos que intervienen en la simulación de procesos son dinámicos, discretos y estocásticos y describen y analizan las características operacionales del sistema real que representan. Esto permite responder a ciertos interrogantes para apoyar su diseño y optimización.

2.4.2 Simulación

La simulación de sistemas por ordenador está basada en una generalización del concepto de experimentación del método científico, según el cual en lugar de realizar los experimentos sobre el sistema real, se realizan sobre un modelo dinámico que lo representa, de manera que si el modelo es una representación válida del sistema entonces los resultados de la experimentación con el modelo pueden transferirse al propio sistema. En esta misma línea R.E. Shannon [Shannon, 75] concluye: "Simulación es el proceso de diseñar un modelo de un sistema real y llevar a cabo experiencias con el mismo con la finalidad de comprender el comportamiento del sistema o de evaluar nuevas estrategias (dentro de los límites impuestos por un criterio o conjunto de ellos), para el funcionamiento del sistema."

Es un hecho admitido que las altas demandas del mercado imposibilitan el uso intensivo de la experimentación como herramienta de evaluación. Por ello las tecnologías de simulación numérica son hoy día un complemento indispensable ya que permiten minimizar actividades costosas en tiempo y dinero, disminuir el riesgo de las decisiones tempranas y explorar un campo más amplio de posibles soluciones.

La simulación puede intervenir en cualquiera de las fases del proceso de diseño del sistema, tanto en la concepción del mismo, como en su diseño preliminar y consiguiente estudio de viabilidad, en el diseño de detalle y en la fase de construcción para proceder a evaluaciones y asesoramientos, o en la fase de utilización y mantenimiento para poder evaluar escenarios alternativos y encontrar respuestas a preguntas del tipo "que pasaría si".

Los entornos de simulación pueden ser de muy diversa índole y pueden llegar a ser muy complejos (meta-entornos). Sus objetivos principales como son la predicción, el entrenamiento, la resolución o el aprendizaje.

La simulación es una disciplina que cobró relativa importancia con la aparición de una técnica numérica (Método de Montecarlo) durante la segunda guerra mundial de manos de Von Neumann y Ulan [Eckhard, 87] y cobró especial importancia en Ingeniería Industrial a partir de la década de los 60. La gran evolución de los métodos informáticos tanto en su aspecto hardware como software, ha permitido afrontar la resolución de complejos problemas físicos matemáticos cuya resolución analítica resultaría prácticamente imposible. La simulación intenta reproducir la realidad a partir de la resolución numérica mediante ordenador de las ecuaciones matemáticas que describen dicha realidad. Por lo tanto hay que asumir que la simulación es tan exacta como las ecuaciones de partida y la capacidad de los ordenadores para resolverlas, lo cual fija límites a su utilización.

La introducción del CAD dio lugar a un gran avance en la etapa del diseño conceptual de nuevos productos. Por contra, el CAE (Ingeniería Asistida por Ordenador) se encuentra en una etapa de mucho más primaria. Sin embargo la verdadera reducción del bucle diseño-desarrollo se produce cuando ambas técnicas actúan conjuntamente. La primera para definir el producto y la segunda para simular su comportamiento en las condiciones de servicio. Sólo la conjunción de ambas técnicas hace posible diseñar para hacer realidad los tres grandes objetivos del diseño moderno: conseguir una fabricación a un costo competitivo, en función de la utilización real en servicio y casi al primer intento. Mediante la simulación numérica y técnicas CAD/CAE es posible generar

sólidos de aspectos casi reales, comprobar su comportamiento bajo diversas condiciones de trabajo, estudiar el movimiento conjunto de grupos de sólidos, etc. Esto permite un conocimiento mucho más profundo de un producto antes de que exista físicamente, siendo posible detectar muchos de los problemas que de otro modo se hubieran detectado en el servicio real.

Actualmente el uso de la simulación está ampliamente extendido en campos como los sistemas de ordenadores, telecomunicaciones, transporte, energía, aplicaciones militares y navales, manufactura, servicios y logística.

2.4.2.1 Simulación vs. soluciones analíticas

El dilema modelos analíticos frente a modelos de simulación debe resolverse en cada caso ateniéndose al tiempo de sistema, objetivos del estudio, características del modelo, los costes, etc. Modelos de simulación y modelos analíticos no deben considerarse siempre como antitéticos, en muchos casos pueden jugar un papel complementario sirviendo la simulación para verificar la validez de las hipótesis para el modelo analítico, o el modelo analítico para sugerir cuáles son las alternativas razonables que hay que investigar por simulación.

Inicialmente los sistemas de ecuaciones diferenciales, han sido las herramientas más utilizadas para modelar sistemas. Para resolverlos, se han reducido muchas veces a sistemas lineales, siendo el álgebra vectorial, el análisis funcional y la teoría general de operadores herramientas útiles que se han mostrado eficaces en el estudio de la física, economía, biología, sociología, etc. Sin embargo, ya en el siglo XIX este impresionante arsenal matemático, base de toda la ciencia clásica, empezó a mostrar su insuficiencia ante dos circunstancias que afectan al estudio de los sistemas: la incertidumbre y la imprecisión. Para la primera el Cálculo de Probabilidades y la Estadística con sus diversas ramas han dado solución a muchos problemas, pero su correcta aplicación está sujeta a fuertes restricciones teóricas y prácticas. Asimismo, el modelado analítico de sistemas puede resultar inviable cuando se quiere introducir sincronización en las acciones y reacciones entre partes del modelo así como cuando se quieren introducir reglas o lógica de comportamiento. Además, la cada vez mayor complejidad de los problemas y el estudio de sistemas abiertos hacen que la formalización matemática clásica sea insuficiente.

Así han surgido otros artificios, como la caja negra, procedente de la teoría de regulación automática (cibernética y robótica), la topología de redes, la algorítmica y las técnicas de la matemática discreta, la programación matemática (lineal, cuadrática, dinámica) y todas aquellas técnicas que conforman la Investigación Operativa o Investigación de Operaciones (problemas de transporte, secuenciación óptima de actividades y tareas, la teoría de la decisión y de los juegos, etc.)

Es obvio señalar que la utilización de estas herramientas se ha visto potenciado por el uso generalizado del ordenador. El ordenador ha permitido también el tratamiento de problemas para los que la formalización matemática del sistema, tanto en sus elementos deterministas como en los aleatorios, es muy difícil o imposible. Esta dificultad se ha reducido en gran medida gracias al empleo cada vez más generalizado de las llamadas técnicas de simulación, las cuales permiten el modelado y estudio de muchos sistemas de estructura compleja, obteniendo soluciones numéricas aproximadas.

El proceso de construcción de un modelo de simulación por ordenador representa el grado de conocimiento que se tiene del sistema en cuestión. Además, el modelo debe entenderse como un instrumento de investigación sometido a revisión continua de refinamiento progresivo en la comprensión del sistema. El proceso no es secuencial, sino iterativo.

La construcción de un modelo de simulación ha pasado, de ser una labor reservada a especialistas en programación de difícil y costosa realización a ser un ejercicio estructurado alrededor de la utilización de entornos cada vez más amables y flexibles que permiten aprovechar la característica más destacable de la simulación: la posibilidad de estudiar la evolución dinámica de los sistemas a lo largo del tiempo.

Los procedimientos de simulación son herramientas de trabajo adaptadas a las necesidades de cada etapa del proceso de diseño. De este modo la simulación numérica puede ser utilizada por los ingenieros de diseño desde el inicio del proyecto, en orden a sintetizar soluciones óptimas para los nuevos productos. Dichas soluciones óptimas deben tener en cuenta los requisitos iniciales y aprovecharse de la experiencia anterior a través de los procedimientos validados de análisis mediante simulación. Con este enfoque se consigue que toda la experiencia empírica existente quede incorporada a los procedimientos de simulación y que el uso sistemático de estos procedimientos en la etapa de diseño garantice el máximo aprovechamiento de todo el conocimiento disponible.

Es importante insistir en la diferencia entre validación, entendida como la demostración de la adecuación de la simulación para una finalidad concreta y la exactitud entendida como la medida del grado de correspondencia entre simulación y realidad. En este sentido es posible encontrar simulaciones que ofrecen resultados válidos para un propósito aún con bajos niveles de exactitud, y viceversa, procesos de simulación capaces de ofrecer altos niveles de exactitud y que, sin embargo, no son válidos para el uso por su alta complejidad, por la dificultad de su aplicación o por exigir excesivos recursos de cálculo.

La masiva utilización de la informática en la enseñanza y en el entorno industrial, la sorprendente y revolucionaria evolución de los ordenadores personales en cuanto a tamaño, coste, velocidad, software, etc. han ayudado sin lugar a dudas a que la simulación digital o simulación por ordenador sea hoy en día la herramienta más utilizada para realizar experimentos de simulación de sistemas.

2.4.2.2 Tipos de simulación

En función del uso de la variable tiempo, las técnicas de simulación convencional se dividen en cuatro grupos:

- Simulación de sistemas continuos

El tiempo es intrínsecamente una variable continua. Así, para realizar una simulación real de un proceso se debería manejar un tiempo que variase de forma completamente continua, lo que únicamente es posible en la simulación analógica. La primera aproximación para introducir el tiempo desde una simulación digital es tomar como reloj un contador con incrementos constantes, lo que produce un reloj síncrono. A medida que los incrementos disminuyan, la aproximación al sistema físico mejorará, a costa de un mayor tiempo de cálculo.

- Simulación de procesos por lotes

Existe otra serie de procesos, denominados por lotes, en los cuales no se precisa una evolución continua del tiempo, ya que las modificaciones que existen en el sistema a simular se produce cada ciertos intervalos de tiempo y no uniformemente espaciados en el tiempo. En este caso se precisa un reloj asíncrono, cuya actualización depende de los eventos producidos en el sistema.

- Simulación combinada

La simulación combinada contempla procesos mixtos en los que existen subprocesos de tipo continuo, junto con subprocesos en lotes. En este caso el tiempo se trata como un reloj síncrono, pero teniendo en cuenta los eventos que se producen de forma asíncrona.

- Simulación cualitativa

Es uno de los tipos de simulación de mayor novedad en la que se utilizan modelos simbólicos de sistemas continuos que son capaces de predecir la evolución de los sistemas. En este tipo de simulación no se manejan valores numéricos del tiempo, utilizándose los conceptos de instantes e intervalos de tiempo entre los que se puede fijar únicamente una relación de orden (anterior-posterior).

- Simulación de Monte Carlo

Por último, existen una serie de sistemas en los que el factor tiempo no influye significativamente, produciéndose las técnicas de simulación mediante el método de Monte Carlo.

2.4.2.3 Aplicaciones y limitaciones

Los modelos necesarios para la realización de experimentos de simulación se usan para el estudio y la optimización del rendimiento de sistemas reales. Sin embargo, la clave de su éxito depende en gran medida de su correcta aplicación. La simulación es el procedimiento más adecuado en muchos casos y la única alternativa tecnológica en muchos otros. Esto resulta especialmente obvio en aquellos casos en los que las características del sistema que se pretenden estudiar hacen inviable, por razones físicas o de coste, la experimentación directa sobre el sistema. El mundo de la producción industrial, del tráfico, la aeronáutica, la industria del automóvil, etc. Son claros ejemplos de esta situación. Incluso en aquellos casos en los que es posible la experimentación directa la simulación puede ofrecer ventajas tales como un coste inferior, tiempo, repeticiones y seguridad. Finalmente, es frecuente que los experimentos persigan el objetivo de determinar la respuesta del sistema en condiciones extremas, lo que puede resultar peligroso e incluso ilegal en la vida real. Del mismo modo se recurre a la simulación cuando no existe un modelo matemático, o cuando existiendo, no se han desarrollado métodos analíticos para resolverlo y cuando la resolución de estos métodos es muy compleja.

La simulación de procesos constituye un método eficiente de ensayo no destructivo rápido, barato y flexible para la predicción del impacto, valor y costo de los cambios propuestos. Algunas veces es la única opción para estudiar un sistema real que, por ejemplo, no existe, es susceptible de ser destruido o su estudio supone una gran inversión. Además, permite el modelado de sistemas complejos con variabilidad e incertidumbre y posibilita la integración de operaciones de distintos departamentos en el análisis global.

En resumen y como regla general la simulación de procesos, es apropiada cuando:

- El desarrollo de un modelo estocástico es muy difícil o imposible (demanda de artículos, tiempos de espera, etc.).
- El sistema tiene una o más variables aleatorias relacionadas.
- La dinámica del sistema es extremadamente compleja: problemas matemáticos determinísticos irresolubles por medio de técnicas analíticas (fenómenos complejos de espera y asignación de tareas, etc.), o sistemas imposibles de describir en términos de ecuaciones matemáticas (economía de un país).
- Se requiere del estudio de su comportamiento en un período de tiempo (producto nacional bruto de un país o ventas anuales de una compañía, etc.).
- Es necesario realizar experimentos de validación imposibles o sumamente costosos (ingeniería aeroespacial, etc.).

A pesar de su utilidad la simulación no puede considerarse como una panacea capaz de resolver todo tipo de situaciones, aún contando con la ayuda de los lenguajes especializados para la simulación, o de los avances que han representado los entornos de software específico. La realización de un estudio de simulación puede comportar un esfuerzo y un consumo de recursos no despreciable en cualquiera de sus fases: definición del problema, recogida de información, construcción del modelo y programación del mismo, realización de los experimentos de simulación por ordenador, validación del modelo y de sus resultados, etc. Sistemas complejos pueden conducir a programas largos y complejos que requieran cantidades importantes de recursos computacionales.

Esto es así debido a que la mayor parte de los procesos a simular no son, en general, simples y el manejo de gran número de variables exige un mayor cuidado y profundidad en el estudio de la propagación de errores. Por lo tanto, el control del “método” y del “software” utilizado en la resolución de un problema adquiere un peso crucial a la hora de evitar la obtención de conclusiones erróneas.

Además la simulación no resuelve los problemas por sí sola sino que ayuda a identificar los problemas relevantes y a evaluar cuantitativamente las soluciones alternativas. Algunas veces da un falso sentido de seguridad y algunas soluciones son pasadas por alto y otras menos buenas son consideradas como óptimas. Esto es así debido a que es difícil interpretar y verificar los resultados de la simulación cuando se trata con sistemas de excesiva complejidad que no se conocen en profundidad.

2.4.3 Tipo de herramientas computacionales

Hay un gran número de aplicaciones que de uno u otro modo automatizan parte del proceso de diseño. Actualmente, para casi cualquier proceso de fabricación se dispone de herramientas informáticas que soportan fases de este proceso. No obstante, los tres campos clásicos de aplicación son la ingeniería civil, el diseño industrial y el diseño de hardware. Este tipo de herramientas se clasifican en tres tipos principales: herramientas de dibujo y visualización, herramientas de análisis de propósito general o específico y sistemas expertos.

2.4.3.1 Herramientas de dibujo y modelado de sólidos

El modelado de sólidos permite visualizar en 3D cada una de sus partes, entender las relaciones geométricas y funcionales existentes entre sus componentes y ayudar al proceso de diseño en la evaluación y la documentación del producto.

La gran mayoría de las herramientas CAD que existen en el mercado son herramientas de dibujo. El 90% de los usuarios de paquetes CAD son diseñadores, no ingenieros de diseño.

2.4.3.2 Herramientas de análisis

A la hora de elegir una herramienta de propósito general o específico hay que tener en cuenta la velocidad de ejecución, los recursos necesarios, la interfaz y la integrabilidad con otros programas.

2.4.3.2.1 Herramientas de análisis de propósito general

Las herramientas de análisis de propósito general son de dominio independiente y permiten evaluar cualquier cosa que se pueda modelar en términos de ecuaciones matemáticas simples.

Las hojas de cálculo disponibles desde los finales de los años setenta y que almacenan y realizan cálculos y gráficos de datos son un tipo de estas herramientas. Durante el proceso de diseño se usan frecuentemente en el análisis para exploración de la sensibilidad de parámetros en función de las variaciones de los demás. A veces presentan cierta utilidad en el diseño de matrices de decisión y en la planificación de actividades. Sin embargo, tienen grandes limitaciones en el caso de problemas de mayor complejidad.

Este tipo de problemas más complejos pueden afrontarse mediante software para la resolución de ecuaciones numéricas y simbólicas. En los programas para resolver ecuaciones numéricas se crea una matriz, se realizan cálculos y se representan gráficamente los resultados. Por otra parte los programas para resolver ecuaciones simbólicas son mucho más potentes y consideran cada variable como un objeto con una relación conocida con otras variables. En el mercado se encuentran disponibles en gran cantidad paquetes de este tipo.

Además de las hojas de cálculo y del software para la resolución de ecuaciones simbólicas y numéricas, a finales de los 80 comienzan a estar disponibles las herramientas de diseño paramétrico y variacional. Las herramientas de diseño paramétrico son sistemas orientados a la geometría del producto de tal modo que todo cambio dimensional se propaga a través del dibujo. Son muy usadas durante el proceso de diseño, pues permiten rápidos cambios geométricos. Por otra parte, las herramientas de diseño variacional son combinaciones de una herramienta de diseño paramétrico y un programa de resolución de ecuaciones simbólicas.

Otro tipo de herramienta de análisis de propósito general son los lenguajes de programación de alto nivel tales como C++, Visual Basic, Matlab, etc. que alejan al programador de las tareas de bajo nivel del ordenador (lenguaje máquina y ensamblador) y suelen ir apoyados en un conjunto de librerías que facilitan en gran medida la tarea de modelado de los sistemas y reducen el tiempo de ejecución del programa. El mayor inconveniente de este tipo de programación es que se requiere un alto grado de especialización.

2.4.3.2.2 Herramientas de análisis de propósito especial

Las herramientas de propósito general, resuelven ecuaciones sin importar lo que representen. Las herramientas de propósito especial pueden ser aplicadas únicamente a campos específicos como el análisis de esfuerzos y deformaciones, el análisis cinemático y dinámico y el análisis térmico y de fluidos. La resolución de ecuaciones de este tipo puede realizarse por métodos clásicos o numéricos. Los métodos clásicos proveen un análisis rápido para problemas de tipo estándar. Esto es una desventaja puesto que si el análisis se hace más complejo es necesario hacer uso de otro tipo de métodos como son los numéricos (por ejemplo, Elementos Finitos). Dentro de este tipo de herramientas destacan los lenguajes específicos y el de software específico de simulación y las herramientas CAD/CAE/CAM (CAM: Manufactura Asistida por Ordenador).

Los lenguajes específicos de simulación son similares a los lenguajes de programación de alto nivel pero están especialmente preparados para determinadas aplicaciones de simulación. Así, suelen venir acompañados de una metodología de programación apoyada por un sistema de símbolos propios, por ejemplo diagramas de flujo, que simplifican notablemente el modelado y facilitan la posterior depuración del modelo.

Entre estos lenguajes específicos podemos nombrar los siguientes: MIDAS, DYSAC, DSL, GASP, MIMIC, DYNAMO, GPSS, SIMULA, CSSL, CSMP, ACSL, DARE-P, DARE-Interactive, C-Simscript, SLAM, SIMAN, SIMNON, SIMSCRIPT-II-5, ADA, GASP IV, SDL, etc. Muchos de estos lenguajes dependen fuertemente de los lenguajes de propósito general como es el caso de Slam o Siman que dependen de Fortran para las subrutinas. La limitación de un lenguaje específico de simulación está en que no permite

desarrollar más allá de para lo que está diseñado, pero como contrapartida el usuario sólo precisa disponer de los conocimientos de programación relativos al producto.

El software de simulación proporciona un marco de trabajo natural para el uso de modelos de simulación y la consecuente reducción significativa del esfuerzo necesario para su programación. Los bloques básicos de construcción del lenguaje son mucho más afines a los propósitos de la simulación que los de un lenguaje de tipo general. Además, los paquetes de software especialmente diseñados para simulación facilitan al simulador las tareas de comunicaciones, la depuración de errores, la generación de escenarios, la manipulación "on-line" de los modelos, etc.

Otro concepto importante es el de la Simulación Visual Interactiva, que puede definirse como aquella que posibilita la creación gráfica de modelos de simulación y permite mostrarlos por pantalla dinámicamente mediante la interacción existente entre el usuario y el programa en ejecución. Los productos de modelado visual permiten realizar prototipos en tiempos muy reducidos. SIMFACTORY, WITNESS, FACTOR/AIM y FIX DEMACS entre otros, son productos de simulación para la resolución de problemas en el ámbito de la producción y son ejecutables sobre ordenadores personales o Workstations. Éstos permiten construir modelos complejos de manera incremental a partir de la selección de entidades predefinidas.

La evolución de los ordenadores y del software comercial se dirige hacia sistemas que puedan ser manejados por personas no-especialistas, con plataformas de cálculo cada vez más potentes a menor coste. Las técnicas orientadas a objetos conducen a programas de utilización más intuitiva. Todo ello nos sugiere un incremento considerable de la aplicación de las técnicas de simulación. Sin embargo, a pesar de todo, se estima que en el mercado sólo se tiene en cuenta la aplicación de técnicas de simulación en menos de la mitad de los casos en los que podría aplicarse.

El software comercial existente en el mercado actual puede agruparse en varias gamas de simuladores. Los simuladores de sistemas de producción y logística (GI-7, Modsim, Siman, Simfactory, Taylor, Witness y Factor/Aim) permiten la simulación de sistemas de colas, transporte, etc. mediante elementos básicos tales como piezas, procesos, máquinas, *buffers* y transporte entre otros. Este tipo de simulación es en general dinámica, estocástica y discreta (aunque pueden simularse procesos continuos). También existen simulaciones para sistemas mecánicos que determinan posiciones, velocidades y presiones de cilindros hidráulicos, actuadores, etc. Se trata de simulaciones dinámicas, deterministas y en tiempo continuo.

Otro tipo de simuladores son los paquetes de cálculo de elementos finitos (que acostumbran a ser módulos de sistemas de CAD-3D) y los Sistemas Expertos que se basan en la Inteligencia Artificial para poner a disposición del diseñador un experto representado en términos de reglas para la síntesis de programas automáticos de diseño. Los sistemas expertos han sido satisfactoriamente aplicados en la configuración del diseño, selección del diseño, estimación de costos y planificación del proyecto. Desgraciadamente los problemas de diseño complicados o significativamente originales han demostrado no estar a su alcance.

Como ya se ha indicado, el proceso de diseño se puede dividir en una etapa de síntesis, en la que se crea el producto y una de análisis donde se verifica, optimiza y evalúa el producto. Las herramientas de automatización del diseño asisten al desarrollo y evaluación de múltiples alternativas de diseño. Las técnicas de CAD/CAM/CAE son

herramientas de propósito especial que incluyen el Diseño Asistido por Ordenador (CAD), la Manufactura Asistida por Ordenador (CAM), la Ingeniería Asistida por Ordenador (CAE), el análisis de Elementos Finitos, la Tecnología de Grupo (GT) y Planificación de Procesos Asistido por Ordenador (CAPP). La aplicación de las mismas consigue abaratar costes, aumentar la calidad y reducir el tiempo de diseño y producción.

Una herramienta CAD/CAM/CAE se compone de un sistema de modelado geométrico, una interfaz de usuario, una base de datos y un sistema de simulación fundamentado en métodos numéricos para la evaluación del diseño en el cumplimiento de requerimientos funcionales, previo a su manufactura.

La existencia de multitud de sistemas CAD/CAM/CAE da lugar a problemas de incompatibilidad de lenguajes. Los traductores directos y los indirectos (formatos neutros) permiten solventar la deficiencia de intercambio de información entre diferentes sistemas. Formatos como IGES (Initial Graphics Exchange Specification) y STEP (Standard for the Exchange of Product model data) son estándares de intercambio de datos entre diferentes sistemas CAD/CAM.

Todo sistema CAM (Manufactura Asistida por Ordenador) utiliza la información del modelo (normalmente en base de datos) como referencia para automatizar el proceso de fabricación, incluyendo la planificación y control del proceso, así como del control de máquinas herramientas. Los sistemas CAPP (Planificación de Procesos Asistida por Ordenador), por el contrario, se usan durante el desarrollo del producto para ayudar al diseñador a evaluar la manufacturabilidad del diseño.

Los campos de aplicación más importantes son la Mecánica, Arquitectura e Ingeniería Civil, Sistemas de Información Geográfica y Cartográfica e Ingeniería Eléctrica y Electrónica. El diseño industrial es el campo típico de aplicación y en el que se comercializan más aplicaciones. Se utilizan modelos tridimensionales, con los que se realizan cálculos y simulaciones mecánicas. Entre las aplicaciones comerciales de tipo general cabe destacar CATIA, IDEAS y PRO/ENGINEER. En Ingeniería Civil podemos encontrar aplicaciones en Arquitectura. Y en el diseño de hardware podemos encontrar aplicaciones para el diseño de placas de circuitos impresos y de circuitos integrados. En este último campo es fundamental la realización de simulaciones del comportamiento eléctrico del circuito que se está diseñando. Muchas de estas aplicaciones son 2D, e incluyen conexión con un sistema CAM.

Para concluir este capítulo indicar que a lo largo de su evolución el proceso de diseño se ha ido concretando en una metodología que hace hincapié en la definición e interrelación de sus etapas (definición y análisis de un problema, síntesis y análisis de los diseños y especificación de la solución final) así como en el papel que tradicionalmente desempeña el diseñador humano en el mismo: la toma de decisiones. En este sentido, el criterio de búsqueda de soluciones para la resolución de un problema es función del diseñador exclusivamente y las herramientas computacionales existentes tienen como objetivo el análisis y la visualización de modelos que permiten estudiar el diseño sin realizar experimento alguno en el sistema real reduciendo el coste del proceso en gran medida.

La mayoría de estas herramientas auxilian al proceso de diseño de forma aislada tratando únicamente ciertas etapas sobre todo la de análisis y especificación de soluciones. Este tipo de software además de ser insuficiente a la hora de implementar

enteramente el proceso no trata la fase de búsqueda de soluciones fundamental para la automatización eficiente del proceso. La computación evolutiva surge como nueva herramienta para la implementación de esta etapa intrínsecamente relacionada con la creatividad. De la eficiencia de esta implementación depende también el éxito de la extracción del humano del proceso de diseño y que sólo participará en él cuando sea necesaria su opinión subjetiva sobre la calidad de las soluciones parciales que tratan de resolver el problema que se plantea.

-
-
-

Computación Evolutiva

3 Computación Evolutiva. Algoritmos Evolutivos.

La Computación Evolutiva se define como el estudio de los fundamentos y aplicaciones de ciertas técnicas heurísticas de búsqueda, optimización, aprendizaje, predicción o clasificación basadas en los principios naturales de la evolución.

La Computación Evolutiva tuvo su origen en las décadas de 1950 y 1960 cuando varios científicos como Holland [Holland, 75, 91, 95], Rechenberg [Rechenberg, 64, 73] y Fogel [Fogel, 63] pretendieron compartir los conocimientos propios de la Biología Evolutiva (ciencia que busca entender la dinámica de la vida) y la Ciencia de la Computación para definir una herramienta que permitiese resolver problemas de optimización en ingeniería.

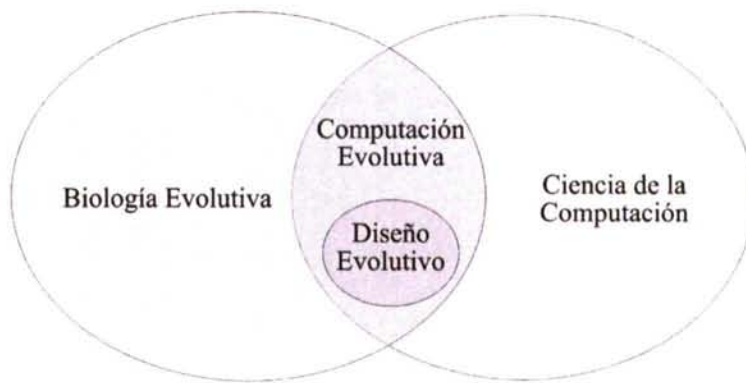


Figura 3.1: Computación Evolutiva

La Computación Evolutiva es una ciencia que se empezó a estudiar hace 30 años. Sin embargo, solamente en los últimos 15 ha habido un número razonable de investigadores trabajando sobre el tema.

Los diseñadores han estado intentando incorporar el uso del ordenador al proceso de diseño desde hace algunos años pero sin demasiado éxito. Se han probado gran cantidad de técnicas para permitir que el ordenador intervenga en la concepción y optimización de diseños, pero solamente en los últimos años se han producido los logros más importantes gracias al crecimiento del uso de técnicas adaptativas de búsqueda.

Los algoritmos de búsqueda definen un problema de diseño donde todas las soluciones posibles al problema pertenecen al espacio de búsqueda y cada punto de ese espacio define una solución [Kanal, 88]. El problema de optimización de un diseño se transforma entonces en el problema de buscar soluciones mejores en el espacio de diseños permitidos. Sin embargo, dada la complejidad de los espacios de búsqueda generados, existe una gran tendencia a encontrar soluciones óptimas no globales. Además en el caso de problemas con varios objetivos, interesa más de una solución. Estos handicaps encuentran solución mediante la aplicación de algoritmos evolutivos.

Estos algoritmos se inspiran en la evolución en la Naturaleza y hacen uso de una analogía con la evolución natural para realizar la búsqueda de soluciones a los problemas. Por lo tanto, en vez de trabajar con una única solución en el espacio de

búsqueda, estos algoritmos consideran una colección grande o población de soluciones candidatas. La flexibilidad, eficiencia y robustez de los técnicas evolutivas [Goldberg, 89] hace que se estén utilizando extensamente para solucionar una amplia gama de problemas. En el dominio del diseño, el uso de la búsqueda evolutiva para optimizar diseños existentes está llegando a ser común.

Está claro que el Diseño Evolutivo en la Naturaleza es capaz de generar diseños increíblemente innovadores. Por lo tanto se espera que el Diseño Evolutivo por ordenador sea también capaz de tal innovación.

En todo Algoritmo Evolutivo se parte de una población inicial de individuos donde cada uno se codifica como un cromosoma y representa una posible solución al problema. La población evoluciona y los individuos se reproducen y mutan de modo que la probabilidad de mantenerse en la población es proporcional a su calidad. Los procesos de reproducción y mutación dan lugar a los diversos tipos de algoritmos evolutivos. La función que evalúa los individuos de la población se denomina función de calidad y determina aquellos que se van a reproducir y eliminar. El hecho de poder seleccionar cualquier función de calidad, hace que los Algoritmos Evolutivos resulten muy potentes cuando no se conocen los detalles del problema, pero sí se conoce el objetivo u objetivos deseados.

Las Estrategias Evolutivas, los Algoritmos Genéticos y la Programación Evolutiva forman la columna vertebral de la Computación Evolutiva. En estos últimos años se ha generado una amplia interacción entre los investigadores de varios métodos de Computación Evolutiva, rompiéndose las fronteras entre Algoritmos Genéticos, Estrategias Evolutivas y Programación Evolutiva. Como consecuencia, en la actualidad, el término Algoritmo Genético se utiliza para designar un concepto mucho más amplio del que concibió Holland.

3.1 Características Algoritmos Evolutivos

Hay quizás tres razones principales por las que consideramos que la opción de los Algoritmos Evolutivos es apropiada para los problemas del diseño: versatilidad, similitud conceptual con el diseño humano y similitud con la evolución natural.

- Versatilidad. Técnica de propósito general.

Los Algoritmos Evolutivos son uno de los muchos tipos de métodos de optimización y búsqueda conocidos en computación como el Descenso de Gradiente, el Recocido Simulado y la Búsqueda Tabú, que han sido aplicados con resultados positivos en ciertas áreas. Actualmente no es posible definir exactamente cuál de estos métodos es el mejor para cada problema o incluso para cada clase de problemas excepto en un sentido muy amplio [Fogel, 95]. Sin embargo, es posible identificar métodos que producen constantemente mejores resultados comparados a los resultados obtenidos por otras técnicas para una amplia gama de problemas. Los Algoritmos Evolutivos pertenecen, de hecho, a esta categoría.

Existen gran cantidad de diseños optimizados con éxito en Computación Evolutiva como, por ejemplo, sistemas de control [Husbands, 96a], composiciones musicales [Horner, 91], [Burton, 97], [Ariza, 02], datos [Radcliffe, 94], [Min, 01], diseños de ingeniería [Bentley, 96, 03], sistemas con tolerancia a fallos [Thompson, 95], [Layzell, 00], juegos [Axelrod, 87, 95], [Jang, 04], aprendizaje [Goldberg, 89], [Holland, 92],

[Smith, 92], [Horn, 94], problemas de orden [Kargupta, 92], [Schaffer, 95], [Knjazew, 03], estrategias de adquisición [Grefenstette, 91], scheduling [Yamada, 95], tratamiento de señales de tráfico [Foy, 92], [Park, 00].

La mayoría de los ingenieros que trabajan en Diseño más que emplear tiempo y dinero desarrollando nuevas técnicas de cómputo especializadas para cada nuevo problema prefieren utilizar un algoritmo robusto y probado con múltiples ensayos, por ejemplo un Algoritmo Evolutivo. De hecho, su gran popularidad ha dado lugar a gran cantidad de congresos, conferencias y libros dedicados enteramente a este tema [Fleming, 01], [Gen, 97], [Bentley, 1998b].

- Similitud conceptual con el diseño humano.

Algunos investigadores [Fogel, 95], [Goldberg, 89], [French, 94] argumentan que la evolución natural y el proceso humano de diseño están directamente relacionados. Está claro que nuestros diseños han evolucionado, del mismo modo que las hachas de piedra de nuestros ancestros se convirtieron en flechas, o como los primeros ordenadores primitivos se han convertido en los actuales ordenadores de gran potencia. El concepto de "selección del más apto" es responsable del aumento de calidad de nuestros diseños y de los seres vivos [Dawkins, 86]. Estudios comparativos de nuestros propios diseños también revelan el desarrollo de "especies" de diseños que se ajustaron a "nichos" claramente definidos [French, 96].

De hecho, Goldberg [Goldberg, 89] procura definir formalmente el diseño humano en términos de evolución de un Algoritmo Genético. Él compara la recombinación del material genético de soluciones padre para formar una nueva solución hijo con un diseñador humano combinando ideas a partir de dos soluciones para formar una nueva solución.

- Similitud con la evolución natural

El debate entre la creación y la evolución se fundamenta en la controversia que suscita el concepto de la vida como fruto de la adaptación y la casualidad, o como un diseño inteligente creado con un objetivo concreto. Richard Dawkins [Dawkins, 86] basándose en la idea de William Paley de que todos los objetos complejos, como puede ser el caso de un reloj, deben tener un diseñador (un relojero), compara la selección natural con un relojero ciego, "...puesto que no ve más allá de sus acciones, no planifica sus consecuencias y no tiene ninguna finalidad prevista. Sin embargo, es prácticamente imposible resistirse a la tentación de considerar que cualquier ser vivo ha sido cuidadosamente planificado para realizar su función". Para Dawkins una fuerza diseñadora materialista y sin propósito, tal como un relojero ciego, sustituye a la deidad "relojera" de la teología natural.

El uso de la evolución para optimizar diseños no es nuevo. Durante tres mil millones de años en un laboratorio de 500 millones de kilómetros cuadrados, se han evolucionado diseños en la Naturaleza que superan con creces cualquier diseño humano en términos de complejidad, funcionamiento y eficacia. Todo ser vivo es una maravilla del Diseño Evolutivo especializado en sobrevivir y reproducirse. Esto es así porque, según la teoría del gen egoísta, los seres vivos son portadores de patrones de información genética en busca de un perfeccionamiento cada vez mayor.

French describió en [French, 94] cómo el árbol exhibe un número de soluciones elegantes de diseño: los árboles presentan resistencia a tensión en el exterior y a compresión en el interior, en todas las etapas del crecimiento. Una cascada de reacciones químicas conducidas por la radiación del sol proporciona carbohidratos para el “alimento”, el agua es ingeniosamente bombeada desde la tierra mediante fibras, numerosos sistemas de control abren y cierran millones de poros minúsculos en las hojas (que recogen el dióxido de carbono) para prevenir la pérdida del agua, etc.

La evolución natural ha estado creando diseños con éxito desde hace mucho tiempo. Incluso un estudio superficial de los diseños extraordinarios existentes en la Naturaleza debe ser suficiente para percatarse del poder de la evolución.

A medida que los biólogos descubren más información sobre el origen y habilidades de las criaturas que viven a nuestro alrededor se dan cuenta que gran cantidad de diseños que el ser humano creyó originales han existido en la Naturaleza desde hace mucho tiempo. De hecho, muchos de los diseños innovadores que ha desarrollado el ser humano presentan características similares a otros evolucionados por la Naturaleza. Por ejemplo, el velcro que presenta el mismo comportamiento que ciertos tipos de semillas “pegajosas” o la forma de la sección de las alas de avión similar a la de los pájaros.

Según lo observado por Paton [Paton, 94]: “un ejemplo muy bueno de cómo la biología puede inspirar soluciones de la ingeniería es el trabajo del profesor O. H. Schmitt que introdujo el término “biomimetic” (emulando a la biología). Es fascinante ver cómo, siguiendo su tesis sobre la simulación de la acción de un nervio, aparecieron cuatro dispositivos electrónicos de sobra conocidos: circuito disparador de Schmitt, amplificador seguidor de emisor, amplificador diferencial y el intercambiador de calor”.

El Diseño Evolutivo es simplemente un proceso capaz de generar diseños, nunca puede ser llamado diseñador. La dificultad de un diseño no es consecuencia exclusiva de un proceso creativo inteligente, puede ser el resultado de una mejora gradual, inconsciente y sin inteligencia. Lo mismo ocurre en el Diseño Evolutivo por ordenador.

3.2 Técnicas de evolución artificial

Existen diferentes técnicas de modelado de procesos de evolución que se han ido plasmando a lo largo de la historia en distintos algoritmos. Dentro de la Computación Evolutiva destacan:

- Algoritmos Genéticos
- Estrategias Evolutivas
- Programación Evolutiva
- Programas de Evolución
- Programación Genética
- Clasificadores Genéticos
- Algoritmos Miméticos
- Algoritmos Macro evolutivos.

En el siguiente apartado se analizarán los Algoritmos Genéticos, la técnica evolutiva más conocida y de uso más extendido.

3.2.1 Algoritmos Genéticos (AG)

Bagley [Bagley, 67] fue el primero en mencionar el término Algoritmo Genético y el que realizó la primera publicación sobre una aplicación del mismo que consistía en la búsqueda de conjuntos de parámetros en funciones de evaluación de juegos, comparándolos con los algoritmos de correlación para aprendizaje. Sin embargo es John Holland el considerado como creador de los Algoritmos Genéticos que los desarrolló, junto a sus alumnos y colegas colaboradores de la Universidad de Michigan, durante las décadas de 1960 y 1970.

El objetivo inicial que perseguía Holland [Holland, 75] plasmado en su libro "Adaptación en Sistemas Naturales y Artificiales" no era diseñar algoritmos para resolver problemas concretos en contraste con las Estrategias Evolutivas y la Programación Evolutiva, sino estudiar, de un modo formal, el fenómeno de la adaptación tal y como ocurre en la Naturaleza. Además diseñó sistemas artificiales basados en los naturales extrapolando los mecanismos de adaptación natural a los sistemas computacionales. Por eso los Algoritmos Genéticos y sus versiones avanzadas se asemejan a la evolución natural más que la mayoría de los otros métodos.

Este mecanismo de búsqueda adaptativa está basada en el principio darwiniano de reproducción y supervivencia del más apto. Los individuos más válidos tienen más oportunidades de transmitir sus códigos genéticos en las próximas generaciones. Estos códigos genéticos constituyen la identidad de cada individuo y están representados por cromosomas.

Los motivos principales de que la evolución sea una fuente de inspiración es que es un método de búsqueda masivamente paralelo que evalúa y modifica millones de especies simultáneamente en función de muchos criterios que varían a lo largo de las generaciones. Además de la capacidad de innovación, la adaptación es otra de las características importantes a imitar debido a que muchos problemas computacionales (por ejemplo, el control de robots) precisan de un programa adaptativo para continuar con un correcto desempeño ante cambios en el entorno y de las necesidades de los usuarios. La simplicidad de los mecanismos evolutivos naturales hace que sean referencia para sus homólogos artificiales. Las reglas de la evolución, aunque de alto nivel, son simples: las especies evolucionan mediante variaciones aleatorias, recombinación y selección natural, donde el mejor tiende a sobrevivir y reproducirse, propagando así su material genético a posteriores generaciones.

La robustez y adaptabilidad son el tema central en las investigaciones sobre Algoritmos Genéticos. Ambos conceptos tratan sobre el equilibrio necesario entre la eficiencia y la eficacia suficiente para la supervivencia en entornos diferentes. Si un sistema artificial es robusto, se podrán reducir, e incluso eliminar, los costes de rediseño, y si se es capaz de lograr niveles altos de adaptación, los sistemas podrán desarrollar mejor sus funciones y durante más tiempo. Sin embargo, el ser humano lo tiene muy difícil si pretende llegar al grado de robustez y adaptabilidad que la Naturaleza ha conseguido en la evolución de los seres vivos. En [Goldberg, 89], Goldberg compara métodos tradicionales de búsqueda con los Algoritmos Genéticos concluyendo que: "aunque nuestro estudio no ha sido ningún examen exhaustivo, de los innumerables métodos de optimización tradicional, obtenemos una conclusión algo inquietante: los métodos convencionales de búsqueda no son robustos". La opinión general de muchos investigadores parece estar de acuerdo con esto, aunque se discute a menudo

exactamente qué algoritmo daría a mejores resultados para cada problema específico. En general se argumenta que un algoritmo tradicional diseñado específicamente para un problema proporcionará resultados mejores para ese problema que un AG, pero que un AG también proporcionará resultados satisfactorios y conlleva la ventaja de poder aplicarse de forma más general.

Los Algoritmos Genéticos difieren de los métodos tradicionales de búsqueda y optimización, en cuatro cuestiones esenciales [Goldberg, 89]:

- No trabajan directamente con parámetros, funciones o variables de control, como los otros métodos, sino con su codificación (cadenas finitas sobre un determinado alfabeto).
- Buscan una población de puntos, no un único punto. Manteniendo una población de puntos muestrales bien adaptados, se reduce la probabilidad de caer en un máximo o mínimo local.
- Emplean la función objetivo, no necesitan derivadas ni otra información complementaria difícil de obtener. De este modo ganan en eficiencia y en generalidad.
- Se valen de reglas de transición probabilísticas, no deterministas. Los Algoritmos Genéticos usan operadores aleatorios para tratar de buscar los mejores puntos tal como ocurre en la Naturaleza.

3.2.1.1 Fundamentos biológicos

Todos los organismos vivos están compuestos por una o más células, cada una de las cuales está provista a su vez de uno o más cromosomas (cadenas de ADN), que definen el “anteproyecto” del organismo del que forman parte. Un cromosoma se divide conceptualmente, en genes, bloques funcionales de ADN que codifican una determinada proteína. Cada uno de los genes posee la información necesaria para desarrollar los caracteres hereditarios (morfológicos, fisiológicos, de comportamiento, etc.) que se transmiten de padres a hijos. Un alelo es la forma concreta que puede tener un gen y es responsable de determinar los rasgos del individuo, como el color del cabello o los ojos. Mientras el genotipo es el conjunto de todos los alelos de un individuo, el fenotipo es la manifestación concreta de un carácter.

La mayor parte de las especies que habitan nuestro planeta son diploides, es decir, almacenan sus cromosomas por parejas (en caso contrario serían haploides). En el caso del ser humano, cada célula contiene 23 pares de cromosomas. Durante la reproducción sexual se produce una recombinación o cruce en donde se intercambian los genes entre cada par de cromosomas de cada padre, para formar un gameto (un cromosoma único). Posteriormente los gametos de los dos padres se emparejan para constituir un conjunto completo de cromosomas diploides. La descendencia está sujeta a mutaciones, donde se produce un cambio en algún nucleótido (bits elementales de ADN) de padre a hijo, esas modificaciones son el resultado habitual de “errores de copia”. La calidad del organismo se define como la probabilidad de supervivencia (viabilidad) o como función de la descendencia que tenga (fertilidad).

La selección natural (las criaturas más débiles no se reproducen con tanto éxito como las criaturas más fuertes) asegura que, en promedio, se obtengan más criaturas aptas en

cada generación. Según lo descrito previamente, la evolución ha producido diseños de seres vivos de gran aptitud para su supervivencia. Es decir, la evolución optimiza el diseño de criaturas vivas para el problema de la vida. La evolución natural actúa con poblaciones grandes de criaturas que se reproducen para generar una nueva descendencia. Esta nueva población hereda algunas características de sus padres (debido al cruce de los cromosomas heredados) y presenta, además, algunas características completamente nuevas (debido a la mutación aleatoria).

Del mismo modo, en todo Algoritmo Genético (AG) en cada población de soluciones al problema, las soluciones más aptas se reproducen con mayor probabilidad, se generan los descendientes de estos padres aptos usando operadores de cruce y mutación, obteniéndose como resultado una nueva población con soluciones cada vez más aptas [Holland, 75].

En los Algoritmos Genéticos, el término cromosoma se refiere a un candidato para la solución de un problema, que a menudo se codifica como una cadena de bits. La codificación más común de las soluciones es a través de cadenas binarias, aunque se han utilizado también números reales y letras. El primero de estos esquemas ha gozado de mucha popularidad debido a que es el que propuso originalmente Holland, y porque resulta muy sencillo de implementar.

El cromosoma de un Algoritmo Genético se representa como un punto en el espacio de búsqueda. El Algoritmo Genético procesa poblaciones de cromosomas, reemplazando sucesivamente una población por otra. En la mayoría de las aplicaciones de los Algoritmos Genéticos nos encontraremos con individuos haploides, concretamente, de cromosomas únicos.

Los genes son los parámetros codificados que definen un individuo de la población y los valores pueden tomar se conocen como alelos. Cada gen es o un bit o un bloque de bits que codifican un elemento particular del individuo. Un alelo en una cadena de bits de ceros o unos (para alfabetos largos cada lugar o locus puede tener más alelos).

El conjunto de parámetros codificados de un individuo se conoce como genotipo, mientras que la solución que definen los parámetros codificados se conoce como el fenotipo. El genotipo de un individuo en un Algoritmo Genético que emplea cadenas de bits es, simplemente, la configuración de bits del cromosoma de ese individuo.

La analogía entre los AGs y el sistema natural se muestra a través de la siguiente tabla resumen:

Naturaleza	AGs
Cromosoma	Palabra binaria, vector, etc.
Gen	Característica del problema (parámetro)
Alelo	Valor de la característica
Locus	Posición en la palabra
Genotipo	Estructura
Individuo	Solución
Generación	Ciclo

3.2.1.2 Fundamentos matemáticos

3.2.1.2.1 Teorema fundamental

A pesar de que la descripción de los Algoritmos Genéticos como programas es simple, existen muchas cuestiones sin resolver acerca de su funcionamiento y para qué tipo de problemas es aconsejado su uso.

El Teorema fundamental de Holland [Holland, 75] intenta definir una base teórica para la Computación Evolutiva fundamentada en la noción de “esquemas”. Esta teoría explica que los AGs funcionan potenciando y recombinando buenos esquemas o “bloques de construcción” de soluciones de una forma altamente paralela. La idea consiste en que las buenas soluciones están construidas por buenas combinaciones de bits o “bloques de construcción”, que confieren altos valores de aptitud en las cadenas en los que están presentes.

Este teorema caracteriza matemáticamente la evolución en el tiempo de una población sobre la que se aplica un Algoritmo Genético. Para ello se basa en el concepto de esquema (schema) o patrón que es una cadena compuesta de ceros, unos y asteriscos. Cada esquema representa al conjunto de cromosomas que contienen los ceros y los unos en la misma posición en la que se encuentran en el esquema y cualquier valor en la posición de los asteriscos. Por ejemplo el esquema 0^*10 representa el conjunto de cromosomas que incluye exactamente a 0010 y 0110. Por lo tanto, una población de cromosomas es el conjunto de esquemas que la representa y el número de individuos que pertenecen a cada uno. Cada esquema describe un conjunto de soluciones que presentan ciertas similitudes. Con A alelos y L genes el espacio de búsqueda es A^L y $(A+1)^L$ es el número total de esquemas. Si el alfabeto es binario, cada esquema de orden $O(H)$ contiene $2^{L-O(H)}$ cromosomas y cada cromosoma es miembro de 2^{L-1} esquemas.

Teniendo en cuenta que el orden de un esquema es el número fijo de genes que presenta y que cada esquema visto espacialmente representa un hiperplano, el esquema de orden 0 *** es el cubo completo. Los demás esquemas son cortes a ese cubo (hiperplanos). De este modo, podemos pensar que un AG corta a través de diferentes hiperplanos en busca de los mejores esquemas. Las hipótesis (esquemas) compiten entre sí para dar explicación a las aptitudes. Las hipótesis van ganando o perdiendo crédito según sean las aptitudes de los cromosomas que las representan. Si al iterar aparece una hipótesis repetidamente asociada a una alta aptitud, posiblemente sea correcta. Para confirmar las hipótesis se muestrean los hiperplanos más prometedores. Sin embargo explorar sólo los más prometedores supone un riesgo.

Holland concluyó que si un esquema tiene una aptitud k veces superior a la del promedio de la población, debe seleccionarse k veces más que el esquema promedio. Esto es precisamente lo que hace el Algoritmo Genético en el proceso de selección. El número de veces que aparece un esquema en la población crece exponencialmente si el esquema es bueno. Según esta teoría en la evaluación de una población, el AG está implícitamente evaluando una media de la aptitud de todos los esquemas que están presentes en la población, e incrementando o decrementando su representación. Esta evaluación simultánea e implícita de varios esquemas es conocido como paralelismo implícito.

Los operadores de reproducción (cruce y mutación) alteran ese resultado rompiendo los esquemas (disrupción) pero son necesarias para explorar nuevos puntos en el espacio de soluciones y evitar la pérdida de diversidad y la convergencia prematura. Cuanto menor sea la distancia de los esquemas menos probabilidad existe de disrupción.

La principal crítica a esta teoría se fundamenta en que la mutación por sí sola funciona bien en determinados problemas, el cruce uniforme introduce mayor disrupción que el cruce por un punto, pero a veces funciona bien. Además, los alfabetos binarios son mejores porque producen más hiperplanos pero en determinados problemas los alfabetos de mayor cardinalidad son muy útiles.

El AG funciona como algoritmo escalador cuando la población es pequeña y los hiperplanos no están muy bien muestreados. Existe demasiada presión selectiva y sólo se reproducen unos pocos cromosomas. En los demás caso el AG funciona como un muestreador de hiperplanos.

Goldberg usa el teorema de los esquemas de forma probabilística y presenta dos principios básicos para escoger una codificación para un Algoritmo Genético: El principio de los bloques constructivos significativos y el principio del alfabeto mínimo. Los bloques constructivos son esquemas con aptitud mayor que el promedio, cortos y de orden bajo. Son cada uno de los genes cuyo valor está fijo en los mejores esquemas. En la evolución los bloques se preservan y combinan para construir la solución óptima. Esto quiere decir que se deben elegir códigos de modo que los bloques sean cortos, de bajo nivel, relevantes al problema subyacente y, a la vez, relativamente ajenos a los bloques que ocupan otras posiciones, segundo, se debe elegir el alfabeto más pequeño que permita una expresión natural del problema.

Posteriormente se siguió investigando en esta línea. En 1997 Stephens y Waelbroek [Stephens, 97] elaboran una nueva teoría de esquemas que estudia la evolución de la población infinita después de muchas generaciones. Poli y Langdon [Poli, 02] mejoran su trabajo estudiando la relación entre tamaño de una población, la aptitud de los esquemas y la probabilidad de éxito.

3.2.1.2.2 Convergencia

No existen muchos estudios teóricos que garanticen la convergencia de un AG. Para que sea aplicable el teorema de punto fijo de Banach es necesario aceptar una nueva población sólo si se mejora la aptitud promedio. Si se sigue este procedimiento da igual la población inicial de la que se parta. El problema es el potencial peligro de toparse con un bucle infinito si no se mejora la población.

Los resultados experimentales demuestran que para la mayoría de los AGs (inicializados con valores aleatorios), la evolución inicialmente progresa a gran velocidad, mientras se combinan y prueban los elementos de la población. A partir de cierto instante, la población comienza a converger y los individuos se asemejan entre ellos cada vez más [Davis, 91]. El AG estrecha su búsqueda en el espacio de soluciones y reduce el impacto de cualquier cambio hecho por la evolución hasta que la población converge eventualmente a una única solución [Goldberg, 89].

La investigación teórica relacionada con el comportamiento de los AGs para diversos problemas está creciendo rápidamente, con análisis cuidadosos de la transmisión de los

esquemas [De Jong, 75], [Kargupta, 93]. El uso de la función de análisis de Walsh [Goldberg, 91], [Deb, 93] y del análisis de Markov Chain [Horn, 93a], [Mahfoud, 93], ha conducido a la identificación de algunos problemas “deceptivos” y “duros” para los AGs [Deb, 92, 93], [Goldberg, 92a, 92b].

3.2.1.2.3 Exploración y explotación

La eficacia, eficiencia y robustez de las técnicas estocásticas depende en gran medida del compromiso entre la explotación de soluciones conocidas y la exploración del espacio de búsqueda para la localización de otras nuevas.

La exploración aporta un impulso innovador al algoritmo de búsqueda permitiendo que, al moverse éste por todo el espacio de soluciones, se generen soluciones muy distintas entre sí aún cuando la calidad media de las soluciones sea menor. Así se puede obtener la variedad suficiente para evitar que la población permanezca estancada en un máximo local y permitir que sea posible la resolución de problemas cambiantes o con varios máximos. Cuanto más exhaustiva sea la exploración mayor será el tiempo de cálculo necesario y más probabilidades existirán de encontrar mejores soluciones. También es cierto que el rendimiento del Algoritmo Genético disminuye al incrementarse la probabilidad de encontrar individuos de calidad inferior.

La explotación de los mejores individuos de una población equivale a una fuerza conservadora que favorece la aparición de soluciones de alta calidad al moverse en el espacio circundante de una posible solución óptima o combinando partes de distintas soluciones, permitiendo así una convergencia más rápida [Holland, 75], [Goldberg, 89], [Michalewicz, 92]. Si, en lugar de una posible solución óptima, se explotan individuos no tan buenos, el algoritmo convergerá hacia un óptimo local.

De entre las técnicas estocásticas, los Algoritmos Genéticos cuentan con un muy buen balance entre la exploración y explotación. El operador de cruce se encarga de la explotación de posibles soluciones y el operador de mutación de la exploración sobre un área determinada del espacio de soluciones.

3.2.1.3 Algoritmo Genético Simple (AGS)

El algoritmo de Holland constituye aún hoy la referencia de muchos desarrollos teóricos y prácticos relacionados con el tema. Goldberg lo retomó y popularizó con el nombre de Algoritmo Genético Simple (AGS) en el que los códigos genéticos están en binario. El proceso de un AGS consta de los siguientes pasos:

1. Codificación del dominio del problema.
2. Generación de una población aleatoria inicial en el espacio de posibles soluciones codificadas al problema.
3. Evaluación (aplicación de la función de calidad) de cada individuo (posible solución) de la población.
4. Evaluación - normalización de la función de evaluación para conseguir una función de calidad o aptitud (fitness).
5. Selección de dos individuos de la población con una probabilidad proporcional a su calidad. Puede realizarse con reemplazo (un mismo cromosoma puede ser seleccionado para ser padre más de una vez), o no. Los individuos seleccionados forman parte del conjunto de reproducción o “mating pool”.

6. Reproducción - cruce de los códigos de los dos individuos seleccionados para formar dos nuevos individuos.
7. Reproducción - mutación de los bits de cada nuevo individuo en función de una determinada probabilidad. Si se cumple la probabilidad cambiar el valor del bit asociado por su complementario.
8. Reemplazo. Los nuevos individuos reemplazan a sus predecesores.
9. Si no se cumple la condición de terminación volver al punto 3.

Se llama generación a cada iteración del proceso. Normalmente un AG se itera de 50 hasta 500 o más generaciones. Al conjunto de generaciones se denomina ejecución. Debido a la aleatoriedad de los AGs, dos ejecuciones con dos semillas diferentes producirán soluciones diferentes. La condición de terminación puede definirse como un número máximo de generaciones o como la estabilización de la calidad media de la población. Generalmente existen múltiples opciones.

En el momento en que las calidades se estabilizan y no alcanzan el valor esperado, es posible inyectar “diversidad genética” añadiendo aleatoriedad a la población y ,en consecuencia, nuevos individuos.

Aunque hay muchos detalles por concretar, el procedimiento descrito es la base de la mayoría de las aplicaciones de AGs.

3.2.1.4 Implementación de un Algoritmo Genético

La implementación de un Algoritmo Genético en un ordenador requiere de la codificación del espacio de soluciones, un mecanismo de selección, operadores genéticos de cruce y mutación y una función de calidad o aptitud. El éxito de un AG dependerá en gran medida de estos factores.

La codificación de las soluciones candidatas debería cumplir que cada solución pueda ser codificada en un cromosoma (completitud) y que cada cromosoma equivalga a una posible solución (cierre). Además preferiblemente ha de cumplir que todas las soluciones deben estar representadas por el mismo número de cromosomas (uniformidad) y que un pequeño cambio en el genotipo conlleve un pequeño cambio en el fenotipo (localidad).

• Operador de selección

Una vez realizada la codificación, se debe decidir el tipo de selección aplicada. La probabilidad de selección de un cromosoma es directamente proporcional a su aptitud. Los cromosomas seleccionados conforman lo que se denomina conjunto de reproducción o “mating pool”.

El método Rueda-Ruleta fue ideado por Holland y en él existe una relación proporcional entre la probabilidad de selección de un individuo y su aptitud. El “valor esperado” de cada individuo se calcula dividiendo su aptitud entre el promedio de la aptitud de toda la población. La ruleta se divide en tantas partes como el número de individuos de la población y a cada individuo se le asigna un sector de la ruleta proporcional a su aptitud.

Escalamiento Sigma es otro método de selección original de Forrest [Forrest, 85] y recuperado por Goldberg [Goldberg, 89] quien la redenomina “truncamiento sigma”.

En esta estrategia el valor esperado de los individuos es función de su aptitud, la media de la población a la que pertenece y de su desviación típica. Así, al inicio de una ejecución, el valor alto de la desviación típica impedirá que los mejores individuos obtengan más superficie de la ruleta. Hacia el final, la desviación típica será más baja y los individuos más aptos podrán multiplicarse más fácilmente.

Kenneth De Jong en [De Jong, 75] introdujo la selección elitista que fuerza a que en la siguiente generación se mantengan cierto número de los mejores individuos de la anterior. Esto es así porque dichos individuos pueden perderse si no son seleccionados o si se destruyen por acción de los operadores de mutación o cruce. Muchas investigaciones defienden que el elitismo mejora significativamente el comportamiento de los AGs.

La selección por rango propuesta por Baker [Baker, 87] previene la convergencia prematura. Los individuos de la población son clasificados en rango en función de su aptitud. El valor esperado de cada individuo es función de este rango (valor relativo) más que de su aptitud (valor absoluto). Este método trata de disminuir la presión selectiva cuando la variación en la aptitud en la población es elevada y aumentar esa presión cuando la variación es baja.

En la selección por torneo dos individuos se eligen aleatoriamente en la población en función de un n° aleatorio se selecciona el más o menos apto de los dos individuos para ser padre. Es posible elegir el mismo padre varias veces. Los métodos citados anteriormente requieren dos fases, en la primera se calcula la aptitud de un individuo y en la segunda su valor esperado. Incluso en la selección por rango se requiere una ordenación de la población. La selección por torneo es similar a este último tipo en términos referidos a la presión para seleccionar, pero computacionalmente es más eficiente, rápido y fácil de paralelizar.

- Operador de cruce

El Algoritmo Genético planteado por Holland proponía un método para desplazarse, de una población de cromosomas a una nueva población, utilizando un sistema similar a la "selección natural" junto con los operadores de cruce, mutación e inversión inspirados en la genética.

Una vez que se ha producido la inicialización aleatoria de la población comienza el bucle principal del algoritmo en donde cada individuo es evaluado en función de la satisfacción de consecución del objetivo o aptitud.

El operador cruce imita la recombinación biológica entre dos organismos haploides. El cruce consiste, normalmente, como en su correspondiente biológico, en un intercambio de material genético entre dos cromosomas de dos padres haploides (gametos). Una vez que del conjunto de reproducción se seleccionen dos padres al azar, el operador cruce elige una posición entre dos cromosomas y cambia las secuencias antes y después de la misma para crear una nueva descendencia.

En el cruce de punto simple, el punto de cruce es un número aleatorio elegido entre cero y el tamaño de la cadena de bits. Existen problemas a la hora de recombinar diferentes bloques de construcción en diferentes cadenas pues es posible la combinación de todos los posibles esquemas. Otros efectos son el llamado efecto bias posicional y el efecto punto final. El Efecto bias posicional [Eshelman, 89] consiste en que los esquemas con

larga longitud de definición tiene una elevada probabilidad de ser destruidos. Los esquemas creados o destruidos por el cruce dependen de la localización de los bits en el cromosoma. El efecto punto final indica que el segmento intercambiado entre los dos padres siempre contienen el “punto final” de la cadena. Para contrarrestar estos problemas se propusieron operadores adaptativos tales como la inversión u otras formas de realizar el cruce.

El propósito del cruce de dos puntos es reducir el efecto “bias posicional” y el efecto de “punto final”. En este método se seleccionan dos posiciones aleatoriamente y entre ambas se intercambian los valores de la cadena. En este tipo de cruce es más difícil que se rompan esquemas con definición de longitud larga y se combinan más esquemas que el punto simple. El segmento que es intercambiado no tiene por qué ser la parte final de la cadena pero siguen existiendo esquemas que el cruce de dos puntos no puede combinar.

El cruce de parametrización uniforme defendido por algunos investigadores como Spears y De Jong [De Jong, 91] consiste en un intercambio que se realiza en cada posición de la cadena con una determinada probabilidad. Este tipo de cruce no presenta bias posicional. Los esquemas contenidos en diferentes posiciones en los padres pueden potencialmente ser recombinados en los hijos.

- Operador de mutación

Para fomentar la exploración del espacio de soluciones se propone la existencia de un operador de mutación que consiste en cambiar el valor de cada bit de cada cadena en función de si un número generado al azar está por debajo de cierta probabilidad. La probabilidad de mutación típica es desde un 0.1% a un 1% del número de genes.

Según Holland, este operador aporta innovación y variación. Con él se asegura que la población no permanezca estática en una posición particular. Existen estudios que comparan la importancia de la mutación frente al cruce destacando su papel para resolver problemas complejos. Spears [Spears, 93] verificó la idea intuitiva de que mientras la mutación y el cruce tienen habilidad de romper los esquemas existentes, el cruce es más robusto a la hora de construir nuevos esquemas. Mühlenbein [Mühlenbein, 92] por otra parte, argumentó que en muchos casos la estrategia de escalado “hill-climbing” trabaja mejor que un AG con cruce y que el poder de la mutación ha sido infravalorado.

En resumen el cruce, la mutación y la selección son conceptos importantes y lo óptimo es obtener un buen balance de los mismos en función del tipo de problema, la función de aptitud y la codificación.

- Función de calidad

El algoritmo requiere de una función de calidad que asigna una puntuación (calidad, aptitud, potencial o fitness) a cada cromosoma de la población actual. La probabilidad de reproducción de cada individuo de la población es proporcional a este valor.

Las soluciones codificadas en un cromosoma compiten para ver cuál constituye la mejor solución ejerciendo una presión selectiva sobre la población, de forma que sólo los mejor adaptados leguen su material genético a las siguientes generaciones. La diversidad genética se introduce mediante mutaciones y recombinaciones.

Normalmente se define como función de aptitud, calidad o fitness a la función de evaluación normalizada. El objetivo de esta conversión es controlar la diversidad de la población y la presión de la selectiva para evitar superindividuos en las primeras generaciones y que la evolución sea errática en las últimas.

- Reemplazo

Existen gran número de técnicas de reemplazo, pero las más usadas son las políticas generacionales y de estado estacionario. Entre las generacionales destacan el reemplazo inmediato donde los nuevos cromosomas sustituyen a los padres y el reemplazo por inserción donde se eliminan los peores cromosomas de la población antecesora y se sustituyen por los nuevos.

En las políticas de estado estacionario se elimina el conjunto de reproducción y en cada iteración se introduce un nuevo individuo, por reproducción de unos padres seleccionados.

3.2.1.5 Algoritmos Genéticos Avanzados

El AG simple es el preferido de los que intentan teóricamente analizar y predecir el comportamiento de Algoritmos Genéticos en donde el proceso de evaluación y reproducción continúa hasta que emerge una solución satisfactoria o hasta que se completan un número especificado de generaciones. [Holland, 75], [Goldberg, 89], [Davis, 91], [Fogel, 95]. Pero en realidad, los AGs son generalmente más avanzados y sus características comunes incluyen una selección natural más realista (es decir selección automática sin la dirección humana), capacidad de detectar cuando la evolución cesa y superposición de poblaciones o elitismo (donde algunos individuos aptos pueden sobrevivir más de una generación) [Davis, 91]. Los Algoritmos Genéticos Avanzados en lugar de aplicar los operadores determinísticos que se usan normalmente en otras técnicas, se ayudan de operadores probabilísticos.

El uso de AGs en aplicaciones de gran complejidad presenta algunos problemas. El más común es la convergencia prematura donde la población converge a extremos locales no óptimos [Davis, 91]. También existen problemas causados por funciones deceptivas (pequeñas variaciones producen enormes cambios de respuesta), que son, por definición, difíciles de solucionar para la mayoría de los AGs. Además, las funciones con ruido [Goldberg, 92], [Lomborg, 91] y la optimización de criterios múltiples dentro de los AGs pueden causar dificultades [Fonseca, 95]. En una tentativa de superar tales problemas, se están desarrollando nuevos tipos de AGs más avanzados [Deb, 93]. Éstos incluyen:

- AGs paralelos, donde procesadores múltiples se utilizan en paralelo. [Adeli, 94], [Levine, 93, 96], [Moore, 03].
- AGs distribuidos, donde se desarrollan poblaciones múltiples por separado con pocas interacciones entre ellas [Whitley, 90], [Mühlenbein, 92], [Alba, 01].
- AGs con razas y especies, donde la población se segrega en distintas especies separadas [Horn, 93b, 94].
- AGs desordenados, que utilizan técnicas “exóticas” tales como cromosomas de longitud variable y proceso de evolución de dos etapas [Deb, 91].

- AGs multiobjetivo, que permiten la optimización de objetivos múltiples [Schaffer, 85], [Srinivas, 95], [Bentley, 96], [Van Veldhuizen, 00].
- AGs híbridos, donde el AG se combina con algoritmos locales de búsqueda [George, 94], [Radcliffe, 94], [Alabau, 01].
- AGs estructurados, que permiten que parte de los cromosomas sean activados o desactivados usando “genes de control” [Dasgupta, 92], [Parmee, 94], [Wu, 02].

Estos Algoritmos Genéticos Avanzados no necesitan para su aplicación de conocimientos específicos sobre el problema a resolver. Además, son paralelizables, es decir, operan simultáneamente con varias soluciones al mismo tiempo, en lugar de trabajar de forma secuencial como las técnicas tradicionales, resultando sumamente fácil ejecutarlos en las modernas arquitecturas masivamente paralelas. Otra ventaja es su robustez en problemas de optimización donde se busca maximizar una función objetivo pues resultan menos afectados por los máximos locales (falsas soluciones) que las técnicas tradicionales. Su versatilidad trae consigo el inconveniente de la falta de especialización [Fogel, 95].

Sin embargo la aplicación de Algoritmos Genéticos Avanzados no asegura que la población vaya a converger y que la solución obtenida sea la mejor. La convergencia depende del tamaño de la población, el número de generaciones, la definición de la función de calidad, el tipo de operadores genéticos utilizados, la probabilidad de aplicación de los mismos, etc. Para evitar la convergencia prematura de una población se puede aumentar la población, mantener la diversidad de los cromosomas, controlar la presión selectiva, controlar la edad de los individuos, preservar los mejores, etc.

Tampoco se evita que el Algoritmo Genético Avanzado pueda desorientarse “deception” durante la búsqueda de la solución. Esto suele deberse a una deficiente codificación, una pobre función de evaluación o la presencia de epistasia (la aptitud de un cromosoma no esta correlacionada con la de sus genes). También puede existir derivas genéticas “genetic drift” debidas al cambio de los alelos de una población debido al azar (no a la selección). Suele existir en poblaciones pequeñas o en soluciones tipo “aguja en un pajar”.

3.2.2 Estrategias Evolutivas

Rechenberg introdujo en la década de 1960 [Rechenberg, 64, 73] el método de las Estrategias Evolutivas para optimizar parámetros reales de ciertos dispositivos al igual que haría posteriormente Schwefel [Schwefel, 75, 77]. Este método de optimización consistía en evolucionar una población de posibles candidatos a la solución de un problema conocido, utilizando operadores inspirados en la selección natural. Los operadores genéticos de los que hace uso son los mismos que en los Algoritmos Genéticos, pero la diferencia consiste en que el operador que guía el proceso evolutivo es la mutación y no el cruce y su probabilidad o bien viene dado por una regla o se codifica como parte del cromosoma como un gen más. Además dicho valor puede ser el mismo para todos los genes o varios valores para distintos grupos de genes. En las estrategias evolutivas simples, la población está compuesta por un sólo individuo que se muta y sustituye a su progenitor si éste es más apto. Las estrategias evolutivas puras que no utilizan operador de cruce son equivalentes a una población de algoritmos de recocido simulado en paralelo. El campo de las Estrategias Evolutivas se desarrolló de modo independiente al de los Algoritmos Genéticos (aunque recientemente han colaborado conjuntamente).

3.2.3 Programación Evolutiva

La Programación Evolutiva, creada por David Fogel [Fogel, 95], presenta similitudes con las Estrategias Evolutivas pero se centra más en el comportamiento del individuo que en su codificación. Inicialmente se concibió como una alternativa a la Inteligencia Artificial clásica para obtener comportamientos inteligentes de forma automática. La Programación Evolutiva se puede considerar como una variante del Algoritmo Evolutivo en la que el lenguaje de codificación de los cromosomas es de más bajo nivel.

El cromosoma de cada individuo está codificado como un programa en un lenguaje de alto nivel que, en general, describe la construcción de una red neuronal. La mutación presenta gran complejidad y consiste en cambiar unas sentencias por otras, comprobando que no se generan individuos sintáctica o semánticamente incorrectos. Su principal handicap radica en que es necesario definir las operaciones permitidas y la codificación para cada problema a solventar, dando lugar a un lenguaje demasiado específico.

En la Programación Evolutiva los genes codifican un autómata finito o máquina de estados en donde cada gen es codificado como una regla que especifica las condiciones que se deben dar (suceso detectado por los sensores del autómata) para que partiendo de un estado origen, sea posible llegar a un estado destino.

estado _ de _ origen + suceso = estado _ de _ destino

Este tipo de definición de la regla se define como imperativa y tiene grandes similitudes con una red neuronal, es el más sencillo de programar, pero no explica porqué el autómata realiza cierta acción. Podemos interpretar la misma regla como declarativa en donde las acciones del autómata pueden modificar su estado para alcanzar un objetivo definido. Este planteamiento tiene grandes similitudes con un sistema experto y permite explicar porqué el autómata realiza una acción.

estado _ de _ origen + acción = estado _ de _ destino

John Koza destaca de entre los autores que han investigado este campo [Koza, 92].

3.2.4 Programas de Evolución

Michalewicz en [Michalewicz, 94] propuso como nuevos algoritmos de Computación Evolutiva a los Programas de Evolución como un refinamiento de los Algoritmos Genéticos. En estos algoritmos se amplía la codificación binaria empleada por los Algoritmos Genéticos hasta admitir cualquier tipo de codificación más cercana al problema a resolver. Cuando la codificación empleada es continua (números reales), los Programas de Evolución y las Estrategia Evolutivas son similares. Actualmente para designar Algoritmos Evolutivos con cualquier tipo de codificación, se utilizan los términos de Algoritmo Genético y Programa de Evolución indistintamente.

3.2.5 Programación Genética

En Programación Genética los genes son codificados como instrucciones de un lenguaje de programación definidos como expresiones simbólicas. John Koza, [Koza, 90] discípulo de Holland, ideó este tipo de algoritmos en 1990 y patentó varios de sus

algoritmos demostrando que la programación genética puede implementarse en cualquier tipo de lenguaje.

3.2.6 Algoritmos Miméticos

Los Algoritmos Miméticos se fundamentan en la combinación de un Algoritmo Genético especializado en la exploración del espacio de búsqueda, con otro algoritmo experto en su explotación para agilizar la convergencia del algoritmo. Un ejemplo de este concepto es el trabajo de investigación de Natalio Krasnogor [Krasnogor, 03] que combina un algoritmo evolutivo con otro de descenso de gradiente.

3.2.7 Algoritmos Macroevolutivos

Los Algoritmos Macroevolutivos fueron desarrollados y formalizados por Jesús Marín y Ricard Solé. [Solé, 97], [Marín, 99]. En ellos la evolución se produce a más alto nivel, los individuos se denominan especies y los operadores básicos son la selección (eliminación de individuos) y la colonización (creación de individuos). La relación entre especies se representa mediante una matriz de conexión cuyos valores son función de la influencia de una especie sobre las otras. El coeficiente de supervivencia de cada especie es la suma de los elementos de la fila correspondiente de la matriz y determina si la especie está o no extinguida.

De entre los investigadores que han desarrollado su trabajo en este tipo de algoritmos cabe resaltar nombres como los de Box [Box, 57], Friedman [Friedman, 59], Bledsoe [Bledsoe, 61], Bremermann [Bremermann, 62], Reed, Toombs y Baricelli [Reed, 67]. Sin embargo, su trabajo no ha tenido, ni con mucho, el impacto de las estrategias evolutivas, programación evolutiva y los Algoritmos Genéticos. Hay que recordar además a los biólogos evolutivos que han utilizado el ordenador para simular la evolución natural realizando experimentos controlados: Baricelli [Baricelli, 57], Fraser [Fraser, 57], Martin y Coreman [Martin, 60]). De todos modos, habría que esperar hasta que la computación electrónica se desarrollara para poder apreciar la consolidación definitiva de la Computación Evolutiva.

En resumen, la versatilidad y similitud conceptual con el diseño humano hacen de los Algoritmos Evolutivos la herramienta más apropiada para implementar computacionalmente la etapa de búsqueda del proceso de diseño. De entre ellos destacan los Algoritmos Genéticos por su simplicidad, robustez y adaptabilidad.

Diseño Evolutivo por ordenador

4 Diseño Evolutivo por ordenador

La aplicación de la Computación Evolutiva para generar diseños ha tenido lugar de forma diferente durante los últimos años. El objetivo común de todas estas aplicaciones consiste en que el ordenador desarrolle diseños por sí mismo, sin que exista una especificación clara de cómo hacerlo y obteniendo cualidades de autonomía, innovación y creatividad.

Hoy, los ordenadores pueden generar nuevos conceptos de diseño y mejorar su funcionamiento mucho mejor que el diseñador humano más experto. La evolución de diseños por ordenador posibilita su empleo en todas y cada una de las etapas del proceso de diseño. Es decir, el diseño asistido por ordenador se está convirtiendo en diseño por ordenador. Los sistemas de Diseño Evolutivos son herramientas avanzadas de software concebidas para dar auxilio al diseñador y que proporcionan una mejora cualitativamente importante en términos de productividad, calidad, velocidad y reducción de costos en el proceso de diseño. El Diseño Evolutivo puede hacer todo esto gracias a la potencia y bajo costo del ordenador. Sin embargo, aunque el campo del Diseño Evolutivo está demostrando algunos resultados impresionantes, los ordenadores no son completamente autónomos y el esfuerzo necesario para concederle características similares a ésta es grande.

Mediante la aplicación de técnicas evolutivas los diseñadores han perfeccionado sus diseños, los artistas han generado morfologías optimizadas desde el punto de vista estético, los arquitectos han evolucionado nuevos planos constructivos de edificios desde cero, los expertos en computación han desarrollado morfologías y sistemas de control aplicables incluso en vida artificial.

Sin embargo, los trabajos desarrollados en esta área suele ser realizados por investigadores aislados dentro de campos diferentes, y aunque existe cierta comunicación interdisciplinaria, parece que la mayoría de los mismos ignoran la existencia de otras investigaciones de cierta similitud conceptual.

El Diseño Evolutivo abarca cuatro aspectos distintos:

- Diseño Evolutivo para Optimización;
- Diseño Evolutivo Creativo;
- Arte Evolutivo; y
- Formas de Vida Artificiales Evolutivas.

Cada uno de estos tipos de Diseño Evolutivo presenta sus propios objetivos y técnicas. Por ejemplo, los investigadores que han realizado trabajos de Diseño Evolutivo para Optimización desean alcanzar un máximo global, en el mínimo tiempo posible. En contraste, los usuarios del Arte Evolutivo requieren un interminable suministro de nuevas formas sin convergencia alguna para extender al máximo el espacio de búsqueda y luego ser evaluadas por los usuarios mediante una interfaz interactiva.

Como normalmente ocurre en cualquier tipo de sistema de clasificación, todo trabajo de investigación no pertenece exclusivamente a una sola categoría, sino que puede ser incluido en dos o más de ellas. De este modo, se definen cuatro áreas de investigación fruto de la combinación de conocimientos de las cuatro citadas anteriormente:

- Diseño Evolutivo Integral;
- Diseño Evolutivo Creativo basado en la Vida Artificial;
- Vida Artificial Evolutiva Estética; y
- Diseño Evolutivo Estético.

La Figura 4.1 muestra estas áreas de investigación y cómo se relacionan entre ellas.

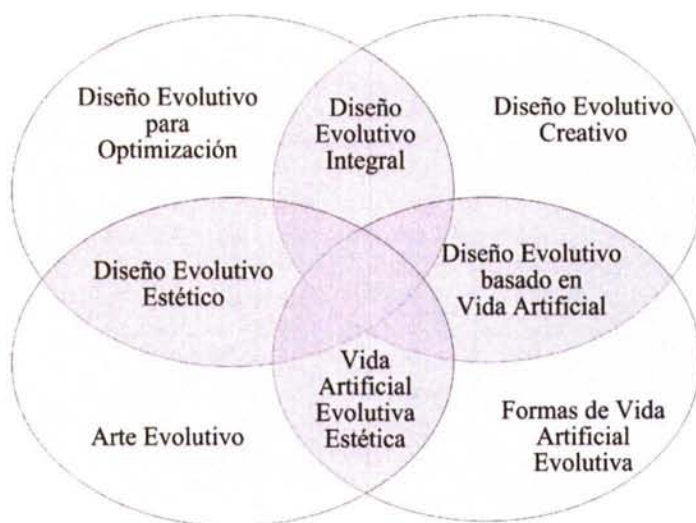


Figura 4.1: Diseño Evolutivo por ordenador

Como ya se ha indicado en el capítulo anterior, aunque el proceso humano de diseño se puede dividir en etapas distintas, generalmente no está realizado explícitamente en estas etapas. En realidad el diseño humano es un proceso evolutivo, con diseños que son optimizados iterativamente. Asimismo, después de repasar la evolución natural, está claro que se puede pensar en la evolución justificadamente como un proceso altamente eficiente de diseño, capaz de inspirar y enseñar a diseñadores humanos.

Finalmente, se han valorado críticamente las tres áreas de investigación más relacionados con el Diseño Evolutivo genérico. La primera de éstas fue la optimización de diseños existentes, demostrando la capacidad del ordenador y de los AGs para optimizar muchos de los diversos tipos de diseño más allá de la capacidad de un diseñador humano. Otra área relacionada es la creación de imágenes artísticas por ordenador que demuestra que los ordenadores pueden crear una amplia gama de imágenes complejas y detalladas cuando son dirigidas por un ser humano. Además, la evolución de las “criaturas artificiales” demuestra que la búsqueda evolutiva puede también crear diseños funcionales sin la interacción humana. Sin embargo, la investigación en la creación automática de diseños todavía está en su infancia, con muy pocos sistemas funcionando con éxito. Por otra parte, casi ningún sistema crea nuevos diseños desde cero (más bien ordenan componentes predefinidos).

Existen grandes paralelismos entre la evolución natural, el diseño humano y la computación evolutiva. El proceso de diseño humano y los Algoritmos Evolutivos de búsqueda desarrollan nuevas soluciones a los problemas usando las mejores características de las soluciones existentes [Goldberg, 89]. Por otra parte, como la búsqueda evolutiva optimiza iterativamente las soluciones, los diseños creados por los

seres humanos son también iterativamente optimizados en el tiempo. Por lo tanto, existe un “triángulo de analogía” entre la evolución natural, el diseño humano y la computación evolutiva, siendo cada uno muy similar y análogo a los otros dos.

El trabajo descrito en esta tesis asimila estos tres procesos. Inspirado por la creatividad asombrosa de la evolución natural, esta investigación procura automatizar el proceso de diseño humano, usando un algoritmo evolutivo de búsqueda para desarrollar nuevos diseños desde cero o partir de alguno existente si es el caso.

4.1 Tipos de Diseño Evolutivo

4.1.1 Diseño Evolutivo para Optimización

El uso de la Computación Evolutiva para la optimización de diseños es una de las primeras aplicaciones de la evolución del diseño. Durante los últimos quince años se han optimizado con éxito una gran variedad de diseños de ingeniería, desde volantes de inercia [Eby, 97] hasta la geometría de un avión [Husbands, 96b].

Aunque el procedimiento exacto usado por los desarrolladores de tales sistemas varía, este tipo de Diseño Evolutivo no se puede clasificar como Diseño Creativo (véase la sección siguiente). Los procesos de Optimización Evolutiva comienzan generalmente el proceso con un diseño existente y parametrizan aquellas partes del diseño susceptibles de ser mejoradas. Estos parámetros se codifican como genes que evolucionan por medio de un algoritmo evolutivo de búsqueda. Los diseños son evaluados a menudo interconectando el sistema a un determinado software de análisis, que se utiliza para evaluar la calidad de cada diseño. Por ejemplo, Goodman [Eby, 97] describe la optimización de volantes usando Algoritmos Genéticos (AGs), en donde se parametrizan y evolucionan las secciones transversales de los volantes de inercia (véase figura 4.2) usando múltiples modelos de Elementos Finitos [Eby, 97]. Keane [Keane, 94], reduce al mínimo la vibración estructural de diseños tales como barras axiales y brazos mecánicos para satélites usando un AG para incrementar al máximo los valores de calidad devueltos por un paquete de Análisis Estadístico de Energía. Gen y Cheng en su libro: “Genetic Algorithms and Engineering Design” [Gen, 97] comenta gran cantidad de ejemplos.

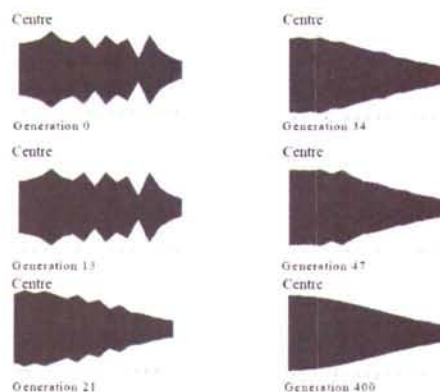


Figura 4.2: Optimización evolutiva de volantes de Goodman

Las representaciones del fenotipo para estos problemas de optimización de diseño son específicas y consisten en diseños existentes cuyos parámetros se evolucionan. A

menudo la transformación del genotipo-fenotipo [Dawkins, 89] es rudimentaria o no existe, simplemente porque no es necesaria para tal evolución. Debido a esto, las representaciones del genotipo pueden estar muy próximas a las representaciones del fenotipo, a menudo con una relación uno a uno entre los genes y los parámetros. Por lo tanto, la adición o la eliminación de genes en el genotipo y los parámetros en el fenotipo, no son realizados generalmente por el algoritmo evolutivo.

Al optimizar diseños, se pone gran énfasis en encontrar una solución tan cerca del óptimo global como sea posible (quizás más que para cualquier otro tipo de Diseño Evolutivo). Debido a esta motivación hacia la optimización global, los investigadores tienden a concentrarse en métodos evolutivos de búsqueda para reducir cualquier tendencia hacia óptimos locales. Además, debido a que el software de análisis usado para proporcionar las funciones de calidad puede tener gran demanda de cómputo, existe a menudo un énfasis fuerte hacia la reducción del número de evaluaciones requeridas antes de que se encuentre una solución final. Se han intentado numerosas técnicas para alcanzar estos objetivos. Para mejorar el funcionamiento, a veces se utilizan representaciones múltiples genéticas en paralelo [Eby, 97]. Se usan muchos tipos complejos de Algoritmos Genéticos [Gen, 97]. Por ejemplo, Husbands [Husbands, 96b] describe el uso de un AG distribuido y de un AG distribuido híbrido con técnicas de pendiente de gradiente para desarrollar la sección de cruce y optimizar así las alas de un avión. Este trabajo demostró que el AG híbrido superó muchos otros algoritmos de búsqueda para este problema.

4.1.1.1 Optimización Específica de Diseños

El desarrollo de sistemas de optimización no genéricos, capaces de optimizar partes explícitamente parametrizadas de diseños existentes, ha sido tratado durante muchos años [Dym, 91], [Adeli, 94]. Los diseños modelados matemáticamente basadas en cálculo como, por ejemplo, los diseños de estructuras tales como puentes o torres de la transmisión se han optimizado según estos métodos [Rozvany, 94]. Éstas y muchas otras técnicas han tratado con la optimización de diseños pero últimamente se ha extendido el uso de AGs y métodos adaptativos similares de búsqueda [Parmee, 94], [Rayward, 95]. Los resultados comparativos entre los AGs y otros algoritmos varían inevitablemente dependiendo de la aplicación, pero para un ejemplo típico de optimización en diseño, el AG supera otros algoritmos tradicionales. Por ejemplo, Tennant [Tennant, 94] compara un AG con recocido simulado y algoritmos de descenso de gradiente para el diseño de absorbentes de microondas, obteniendo como resultado que únicamente el AG es capaz de localizar correctamente el máximo de la función para cada funcionamiento. Husbands encontró resultados similares para la optimización del diseño estructural de las alas de un avión [Husbands, 96b].

Existen numerosos ejemplos de optimización de diseños, la mayoría usan AGs u otros métodos adaptativos de búsqueda como por ejemplo:

Filtros de antenas y radares [Edwards, 98], geometría de aviones [Bouchard, 88], [Oyama, 00], [Husbands, 96b], componentes electrónicos [Gordon, 02], puentes estéticos [Furuta, 95], redes hidráulicas [Savic, 94], materiales absorbentes de microondas [Tennant, 94], satélites [Keane, 96], vibraciones [Moshfreti, 00], servo y micro motores [Hameyer, 96], sistemas para vehículos espaciales [Garipov, 94], topología estructural [Adeli, 94], torres de transmisión [Cai, 96], etc.

Aunque muchas de estas aplicaciones son demostraciones simples y son a menudo implementadas con aparentemente poco conocimiento del Algoritmo Genético usado, se han producido soluciones de optimización con un éxito considerable. Por ejemplo, según lo descrito por Holland [Holland, 92], un Algoritmo Genético es capaz de optimizar tres veces mejor el diseño de la turbina del motor de un jet en sólo dos días, mucho menos de las ocho semanas que necesitaría un ingeniero.

Mientras que una amplia variedad de aplicaciones demuestra el acierto del uso de los ordenadores para evaluar y optimizar con éxito distintos diseños, cada uno de estos sistemas de optimización, sin excepción, sufre de dos desventajas importantes. En primer lugar, cada uno de ellos puede optimizar solamente diseños existentes sin posibilidad de crear nuevos diseños. En segundo lugar, cada uno de estos sistemas son aplicaciones específicas que pueden optimizar solamente un solo tipo de diseños para lo que fueron creados y ningún otro.

4.1.1.2 Optimización Genérica de Diseños

La optimización genérica de diseños (es decir la optimización de más de un tipo de diseño por un solo sistema) es un tema menos común para la investigación. El sistema de Kant "CONFIG" [Kant, 88] permite que sea analizada una variedad de diseños, pero no realiza ninguna optimización. Del mismo modo, "EXPECT" [Libes, 90] permite enlazar múltiples herramientas de análisis, pero no incorpora la optimización. El "Engineer's Associate" de Bouchard [Bouchard, 88] proporciona un marco genérico limitado para trabajar con sistemas que se pueden representar por ecuaciones. El sistema de optimización de propósito general "GPOS" de Culley [Culley, 94] consiste en una toolbox de algoritmos de optimización, capaz de optimizar una gama de diversas aplicaciones (una vez que se han interconectado apropiadamente).

Sin embargo, "Engineous" de Tong [Tong, 92] es quizás el sistema genérico con más éxito, habiendo sido demostrado sobre más de 20 tareas de optimización del diseño, incluyendo la optimización de turbinas, ventiladores, motores de C.C. y fuentes de alimentación. Una gran parte del sistema consiste en software de interconexión complejo para permitir el uso de los paquetes existentes de evaluación del diseño. El sistema confía en gran medida en sistemas expertos con gran cantidad de conocimiento específico para dirigir la evolución de un AG, que ha de cambiarse para cada nueva aplicación. Tong dice que "la versión actual de Engineous ha demostrado el gran impacto que tal sistema puede tener en la productividad y el funcionamiento" [Tong, 92].

4.1.2 Diseño Evolutivo Creativo

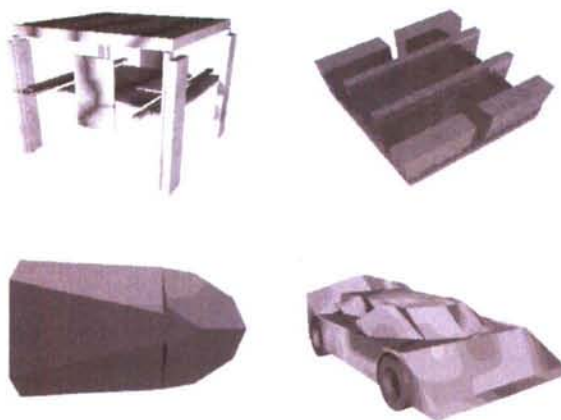


Figura 4.3: Diseño creativo evolutivo de Bentley

Llamar a cualquier cosa generada por ordenador “creativa” es tema de discusión. Para los propósitos del Diseño Evolutivo, ¿qué significa realmente “creativo”?

Escribiendo sobre este mismo tema en “Computers and Creative Design” [Gero, 96], Gero explica que un individuo puede demostrar creatividad cuando diseña y un diseño puede tener características creativas. Gero se concentra en la definición anterior y concluye que un ordenador está diseñando creativamente cuando explora el conjunto de posibles estados del diseño además de explorar espacios individuales del diseño. Es decir Gero indica que evolucionando el número de las variables de decisión además de evolucionar los valores de esas variables, un ordenador está siendo creativo [Gero, 96]. En una línea similar, Boden sugiere en su libro “The Creative Mind” [Boden, 92], que la creatividad es solamente posible yendo más allá de los límites de una representación y encontrando una solución novedosa que no se habría podido definir simplemente por esa representación. Boden, sin embargo, no cree que los ordenadores sean capaces de tal creatividad. Para otros investigadores el Diseño Creativo tiene la capacidad de generar “soluciones innovadoras y sorprendentes”, o son “soluciones novedosas cualitativamente mejores que las soluciones anteriores” [Gero, 96]. Sin embargo, la descripción de Rosenman parece ser la más conveniente: “Cuanto menor sea el conocimiento sobre las relaciones existentes entre las especificaciones y la forma para satisfacer esas especificaciones, más próximo se estará de un problema de Diseño Creativo” [Rosenman, 97].

Por lo tanto, la característica principal que todos los sistemas de Diseño Evolutivo Creativo tienen en común es la capacidad de generar nuevas formas empezando de la nada (es decir de poblaciones iniciales al azar) y ser evolucionadas únicamente por criterios funcionales. Para realizar esto, tales sistemas varían a menudo el número de las variables de decisión durante la evolución [Bentley, 97], [Gero, 96], [Rosenman, 97]. Se pueden generar a menudo sorprendentes soluciones innovadoras, o nuevas soluciones cualitativamente mejores que otras [Bentley, 97], [Harvey, 97]. Si esto significa que estos sistemas realmente están “diseñando creativamente”, o si generan simplemente “diseños creativos”, es otro tema.

La investigación en el campo del Diseño Evolutivo Creativo se refiere a las etapas preliminares del proceso del diseño. Existen dos aproximaciones principales y ambas implican el uso de la Computación Evolutiva para generar enteramente nuevos diseños desde cero, no obstante el nivel en el cual se representan estos diseños es diferente.

4.1.2.1 Diseño Evolutivo Conceptual

En este tipo de Diseño Evolutivo Creativo, las relaciones y las disposiciones de diseños de alto nivel evolucionan interconectando bloques constructivos conceptuales existentes en una tentativa de generar diseños básicos novedosos.

Un buen ejemplo de esto es el trabajo de Pham [Pham, 93], que describe su sistema de diseño básico conocido como TRADES (TRANsmiSSion DESigner). TRADES utiliza un Algoritmo Genético para evolucionar la disposición de un sistema de primitivas constructivas (tales como cremalleras y piñones, engranajes, ...). Cuando se da el tipo de entrada (por ejemplo movimiento rotatorio) y la salida deseada (por ejemplo movimiento linear perpendicular), TRADES genera un sistema de transmisión conceptual para convertir la entrada en la salida. De la misma forma, Chakrabarti [Chakrabarti, 95]. desarrolló un programa capaz de crear diseños simples de dispositivos y máquinas móviles Usando una base de conocimiento y un motor de búsqueda, "FuncSION" podía generar diseños combinando elementos básicos tales como pivotes, barras y palancas. El sistema se ha utilizado hasta ahora para diseñar alternativas a manillas de puertas y como sistema de ayuda en personas con distrofia muscular.

Las representaciones del genotipo y del fenotipo de estos sistemas son a menudo simples, con embriologías rudimentarias, si las hay. Los Algoritmos Evolutivos básicos son generalmente suficientes para este tipo de problema. Sin embargo siempre hay excepciones y el trabajo de Parmee es una de ellas. Parmee [Parmee, 96] describe el uso de AG estructurados para desarrollar un gran sistema hidroeléctrico (éste tendrá lugar en las etapas de oferta de viabilidad del proceso del diseño, después de la etapa del diseño conceptual). Su AG avanzado manipula una jerarquía de diseño de las disposiciones, tipos de presa, longitudes de túnel, modos de operación, etc. y permite que los elementos apropiados sean habilitados o no mediante el control de los genes durante la evolución.

En todos estos sistemas la base del conocimiento determina la operación y las combinaciones de los componentes que se permiten, limitando potencialmente el alcance para diseños conceptuales radicalmente diversos.

4.1.2.2 Diseño Evolutivo Creativo

Según lo descrito previamente, la optimización de diseños existentes es relativamente común, en temas como la creación imágenes artísticas y en vida artificial. Sin embargo, la creación de nuevos diseños es uno de los tipos de Diseño Evolutivo más recientes y está demostrando su gran potencial en un gran número de áreas de aplicación.

El uso de ordenadores para generar formas de diseño en lugar de conceptos de alto nivel predefinidos tiene la ventaja de dar mayor libertad a las mismas. Esta libertad de desarrollar cualquier forma capaz de ser representada y la evolución de tales formas puede dar como resultado conceptos de diseño implícitos [Bentley, 97], [Harvey, 97].

Sin embargo, la dificultad de este tipo de Diseño Evolutivo Creativo es grande, puesto que ella implica a menudo la creación de representaciones dinámicas y rutinas complejas de evaluación.

A menudo este tipo de Diseño Evolutivo apenas implica etapas preliminares de la fase de diseño y hace hincapié en la innovación y la originalidad y no en la producción de soluciones globales óptimas. Las representaciones de formas varían enormemente, pero todas comparten ciertas características. Por ejemplo, debido a que el énfasis está en la generación de nuevas formas, las representaciones del fenotipo son bastante generales, capaces de representar gran número de morfologías alternativas (esto está en contraste con las representaciones para la optimización, que puede definir solamente variaciones de una sola forma). Las representaciones se extienden desde un reparto espacial directo (por ejemplo los voxels), que presentan una transformación genotipo-fenotipo [Baron, 97], a las representaciones altamente indirectas que utilizan los sistemas de reglas o los autómatas celulares (AC) [Coates, 97], [Rosenman, 97]. Por ejemplo, Rosenman [Rosenman, 97] describe el uso de un AG para desarrollar diseños de casas, usando diseños de reglas, Coates [Coates, 97] utiliza la programación genética (PG) para desarrollar formas arquitectónicas abstractas, Schnier y Gero [Schnier, 96] incluso procuran desarrollar nuevas representaciones de alto nivel para la evolución de los planos de una casa en un estilo arquitectónico específico.

Los Algoritmos Genéticos simples son a menudo suficientes para los sistemas de diseño que emplean representaciones sin complejidad [Baron, 97]. Sin embargo, para los sistemas de Diseño Evolutivos Creativos con representaciones avanzadas son necesarios AGs más avanzados. Normalmente, estos AGs se usan para evolucionar la estructura de la representación (por ejemplo número y organización de las reglas) además del detalle (por ejemplo tipo y contenido de las reglas del individuo). Es decir estos AGs son capaces de desarrollar diseños que tienen representaciones de longitud variable (exploran nuevos y diversos espacios de la búsqueda además de los valores de los parámetros dentro de cada espacio [Bentley, 96]). AGs y PG se usan para evolucionar este tipo de formas [Bentley, 97], [Coates, 97], [Rosenman, 97], comenzando generalmente desde formas simples al azar y mejorando gradualmente la estructura y el detalle de estos diseños según criterios funcionales.

El objetivo de este tipo de Diseño Evolutivo no es utilizar el ordenador para generar una única solución óptima global, si no para generar un número de alternativas "creativas". La evaluación de diseños puede ser más difícil, puesto que la mayoría de los paquetes disponibles de análisis se limitan a evaluar tipos específicos de diseños. Por lo tanto, muchos de estos sistemas confían en las rutinas simplificadas de evaluación escritas ad hoc, que pueden analizar todo lo que se les presente, pero quizás no siempre con la exactitud deseada [Bentley, 97].

Bentley [Bentley, 97, 96], describe la evolución, desde cero, de diseños de gran innovación para numerosas aplicaciones, tales como tablas, disipadores de calor, prismas ópticos, formas aerodinámicas e hidrodinámicas, etc., véase fig. 3 (parte derecha superior). Funes [Funes, 97] demuestra el uso de un AG para evolucionar las estructuras de puentes y grúas, que se construyeron posteriormente con modelos LEGO. Un área de aplicación que recibe actualmente mucha atención de los medios es "Hardware Evolutivo", donde se desarrollan nuevos circuitos lógicos y se evalúan usando FPGAs [Harvey, 97]. Ya se ha encontrado alguna solución sorprendente y novedosa usando estas técnicas.

Algunos trabajos inicialmente se concentraron en el área cognitiva de la automatización de diseño creativo [Dyer, 86], [Maher, 89], [Dym, 91], es decir procurando hacer que un ordenador “pensase” de la misma manera que un ser humano, cuando se diseña. Tales sistemas intentaron crear descripciones de diseños en un nivel abstracto, usando típicamente un sistema experto “para diseñar”. Por ejemplo el “EDISON” de Dyer [Dyer, 86] representó dispositivos mecánicos simples como puertas y abrelatas simbólicamente en términos de cinco componentes: partes, relaciones espaciales, conectividad, funcionalidad y procesos. Una combinación de planificación y de invención usando “generalización”, “analogía” y “mutación” trató de modificar estos componentes para satisfacer la especificación del diseño. Desafortunadamente, el nivel abstracto en el cual se realizó el razonamiento fue demasiado bajo, así que el sistema no podía manejar cualquier problema aparte de casos más simples [Pham, 93].

Otra aproximación consistió en la invención basada en “interacciones potenciales de visualización” [Williams, 90]. Esto generó descripciones de diseños en términos de componentes de alto nivel y las interacciones entre ellos, usando el razonamiento cualitativo y álgebra cuantitativa. Una vez más, el sistema propuesto podría ocuparse solamente de diseños altamente simplificados [Pham, 93].

Algunos investigadores anunciaron que habían producido sistemas de diseño preliminar [Ulrich, 87], [Michielssen, 92], [Pham, 93] pero en una inspección más rigurosa, tales sistemas consistían en la optimización de conexiones entre los bloques estructurales existentes de alto nivel o simplemente en la optimización de diseños existentes. Por ejemplo, Michielssen en [Michielssen, 92] describe una aproximación para diseñar filtros ópticos de múltiples capas óptimos que “no requiere de un diseño preliminar”. Sin embargo, el autor describe cómo se usa un AG para determinar “el grosor de las capas del filtro requerido para una respuesta óptima”. Es decir, el AG diseña simples filtros preliminares que consisten en un número fijo de capas cuyo grosor deben ser optimizados.

Otros investigadores favorecen analogías más cercanas con la Naturaleza, usando embriologías artificiales que se asemejan a la natural. Es decir, los genes se utilizan como instrucciones de cómo las formas deben “crecer”, más que para especificar las formas directamente. Por ejemplo, de Garis [de Garis, 92], procuró evolucionar formas simples usando colecciones de “células” con un Algoritmo Genético. Sin embargo, todos los resultados producidos por el sistema fueron decepcionantes, obteniéndose un simple forma en “L” en el caso de mayor complejidad. Para demostrar lo difícil que es evolucionar formas más complejas, la tentativa final fue evolucionar una forma tipo “tortuga” (formada por seis círculos llenos: uno para el cuerpo, otro para la cabeza y cuatro para las piernas). Los autores concluyeron: “la calidad fue del 80% solamente y no parecía una tortuga. Se parecía más a una gota.”. Es posible que su gran complejidad evitara que el sistema fuese escalable y permitiese la creación de diseños más complejos y realistas.

Como ya se indicó anteriormente Rosenman [Rosenman, 96] tuvo más éxito usando una idea similar de evolucionar nuevos planos de casas. Éste es quizás el trabajo que puede ser descrito lo más exactamente posible como creación de diseño, puesto que los diseños reales se generan desde cero. Se evolucionan planos en dos dimensiones usando un AG simplificado para modificar celdas organizadas jerárquicamente usando reglas gramáticas. Esta representación espacial, aunque suena conceptualmente elegante (crecimiento de células para formar diseños), requiere de estructuras enormes y complicadas para definir cualquier diseño compuesto por apenas algunas células. Al

igual que el sistema de de Garis, es posible que la complejidad del sistema evite su escalabilidad y la creación, por tanto, de diseños más realistas.

4.1.3 Arte Evolutivo

El Arte Evolutivo es quizás el tipo de Diseño Evolutivo de más éxito comercial. Aunque la investigación en esta área es menos común que en otros campos, hay quizás más productos de Arte Evolutivo disponibles hoy, que cualquier otro tipo de sistemas de Diseño Evolutivo.

Todos los sistemas de Arte Evolutivo suelen parecerse entre ellos. Todos generan nuevas formas o imágenes desde cero (poblaciones iniciales aleatorias) y todos dependen totalmente de un evaluador humano para calcular la calidad de cada uno de los miembros de la población (basada normalmente en la apariencia estética). Los tamaños de la población son generalmente muy pequeños, para permitir que la generación de calidades de cada generación sea rápida. Los interfaces de usuario son a menudo similares, mostrando los miembros de la población actual en la pantalla en forma de matriz de ventanas, permitiendo que el usuario los evalúe seleccionando una determinada puntuación con un ratón (ver figura 4.4).

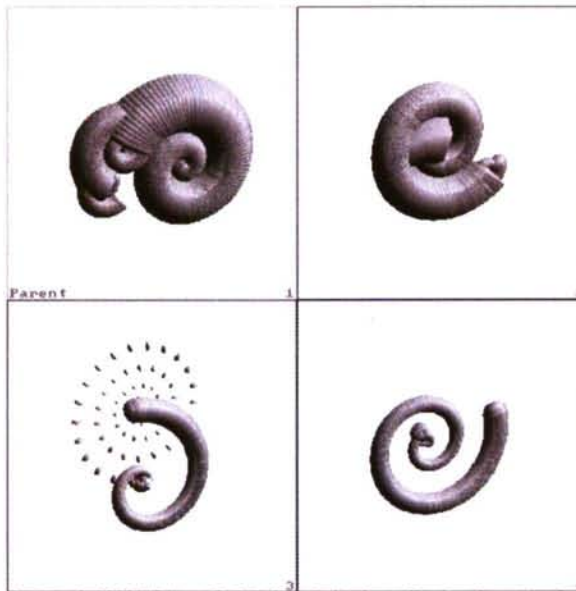


Figura 4.4: Arte evolutivo de Rowbottom

Las diferencias principales entre estos sistemas residen en sus representaciones del fenotipo. Se han empleado una gran variedad de representaciones alternativas, desde ecuaciones fractales (tales como “Interactive Genetic Art” de John Mount [Mount, 98]), a las reglas recursivas usando geometría sólida constructiva [Todd, 92]. Estas representaciones se crean con diversas intenciones. Por ejemplo, las estructuras de árboles recursivos fueron pensadas para asemejarse a las embriologías recursivas encontradas en la Naturaleza, con la esperanza de obtener formas parecidas a las naturales [Dawkins, 86, 89].

Permitiendo que la evolución varíe las estructuras (por ejemplo, cambiar el número de reglas o de formas primitivas), como se hace en el Diseño Evolutivo Creativo, podría aumentar posiblemente la creatividad de tales sistemas.

Un efecto secundario indeseado de muchas de estas representaciones es que generan formas artísticas que tienen estilos muy distintos. Esto puede causar problemas si el artista desea dar crédito a la obra. Limitando el ordenador a un tipo específico de estructura, o un sistema específico de formas primitivas y reglas constructivas, se generará inevitable formas con mucho en común.

Como la evolución está dirigida por una selección humana (es decir la “función de calidad” es el artista), el algoritmo evolutivo no tiene que ser complejo. La evolución se utiliza más como generador continuo que como optimizador. El artista es probable que considere criterios inconsistentes y que cambie de opinión sobre características deseables de la obra durante la evolución, así que la generación continua de nuevas formas basadas en el más apto de la generación anterior es esencial. Por lo tanto, un elemento importante de los Algoritmos Evolutivos usados es la no convergencia. Si las poblaciones de las formas siempre perdiesen diversidad y convergiesen sobre una sola forma, el artista no podría explorar cualquier otra forma. Debido a esto, la mayoría de los sistemas evolutivos del arte no emplean el cruce dentro de sus algoritmos evolutivos. Básicamente se usa el operador mutación, todo descendiente es una copia transformada de sus padres. Esta evolución-mutación es similar a la aproximación que de la que se hace uso en las estrategias de la evolución que se saben que son excelentes para encontrar soluciones a problemas con funciones de calidad que cambian continuamente [Bäck, 96]. Ejemplos de tales sistemas son “Biomorphs” de Dawkins [Dawkins, 86, 89], “Evolutionary Art” de Todd y Latham [Todd, 92] y “Evolved computer graphics” de Sims [Sims, 91].

El uso del ordenador para crear arte (otra vez, a menudo con AGs u otros algoritmos adaptativos similares de búsqueda) está creciendo en popularidad entre algunos artistas. Por ejemplo, Stephen Todd y el artista William Latham [Todd, 92], han desarrollado con éxito muchas imágenes “artísticas” tridimensionales y animaciones. Su sistema utiliza una embriología artificial elegante conocida como la “Form Grow” para permitir la definición de intrincadas formas tridimensionales y texturas. Estas formas se componen de un número de formas primitivas (por ejemplo, espiral, esfera, toro) seleccionadas para dar un distintivo aspecto “biológico” a las imágenes. Un simple algoritmo de búsqueda evolutiva, conocido como “Mutator” (no muy distinto a una estrategia evolutiva (ES), [Rechenberg, 73]) permite la creación y modificación de las formas, dirigida por el usuario del software. Este trabajo ahora ha sido reconstruido por Latham y Atkinson, para crear un producto comercialmente disponible conocido como “Arte Orgánico”.

Un sistema similar fue desarrollado por Husbands, Jermy, McIlhagga e Ives [Husbands, 96b]. Este sistema usa supercuádricas como formas primitivas conjuntamente con un idioma descriptivo recurrente para dichas formas, permitiendo la especificación de una variedad de formas tridimensionales.

Otro ejemplo es el trabajo del ya mencionado John Mount, que muestra su “Arte genético Interactivo”. Él emplea un AG para modificar las ecuaciones fractales que definen imágenes de dos dimensiones. Entonces se invita a los visitantes a que voten lo atractiva que es cada imagen, que proporciona al AG las calidades necesarias para permitir que se produzca una nueva generación de imágenes teóricamente más atractiva.

Todos estos sistemas se puede decir que crean imágenes y formas, puesto que ninguno implica la modificación de imágenes existentes, es decir, todas las imágenes se desarrollan desde el principio al azar (desde cero). Sin embargo, todos estos sistemas requieren que las imágenes sean evolucionadas para ser evaluadas por un ser humano (es decir selección artificial en vez de selección natural). Por otra parte, debido a las representaciones son limitadas, la mayoría de estos sistemas no pueden producir cualquier cosa más que “imágenes bonitas”.

Sin embargo, tales sistemas demuestran la capacidad del ordenador usando búsqueda evolutiva para generar una amplia gama de diversas imágenes bi y tridimensionales. Además, y quizás sea lo más importante, desarrollan una variedad de imágenes altamente originales e inusuales, todos estos sistemas demuestran creatividad por ordenador.

4.1.4 Formas de Vida Artificial Evolutiva

La Computación Evolutiva desempeña un papel significativo en muchos aspectos en el campo de la Ciencia de la Computación conocido como Vida Artificial (VA). Los “cerebros” artificiales, las estrategias del comportamiento, los métodos de comunicación, la resolución distribuida de problemas y otros muchos aspectos se tratan comúnmente usando Algoritmos Genéticos y otras técnicas evolutivas de búsqueda [Cliff, 94].

Toda Forma de Vida Artificial Evolutiva puede ser descrita como un aspecto particular de Diseño Evolutivo. El autómatas celular de Lohn [Lohn, 95] que se evolucionó para ser capaz de replicarse automáticamente, la disposición evolutiva y la estructura de neuronas de Harvey [Harvey, 97] y las morfologías tipo planta y animal de Dawkins y de Sims [Dawkins, 86, 89], [Sims, 94] son ejemplos de este tipo de diseños.

Las motivaciones para la creación de Vida Artificial Evolutiva son generalmente teóricas e intentan imitar los mecanismos de la evolución natural. A menudo, las formas evolucionadas demuestran su enorme potencial para este tipo de Diseño Evolutivo, pero hasta ahora, los usos prácticos siguen siendo escasos. La figura 4.5 muestra algunas de las criaturas virtuales desarrolladas de Sims.

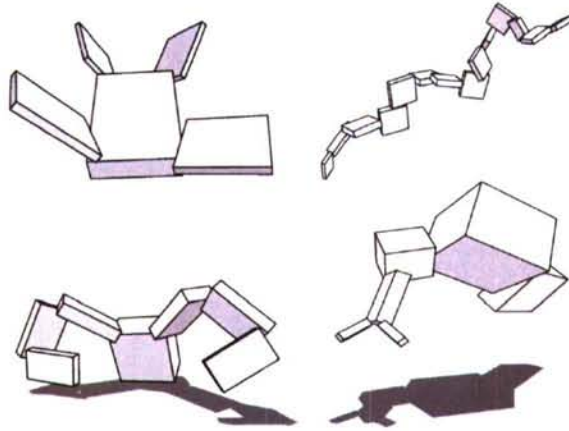


Figura 4.5: Formas de vida artificial evolutiva de Sims

Muchas de las representaciones utilizadas en este tipo de Diseño Evolutivo están inspiradas en la estructura del genotipo de organismos naturales, pero otras son más complejas y específicas [Bentley, 97]. Por ejemplo, Lohn y Reggia [Lohn, 95] utilizan reglas, Sims [Sims, 94] utiliza una estructura jerárquica de cromosoma para definir el “cerebro” y el “cuerpo” y Ventrella [Ventrella, 94] combina parámetros de morfología y control de sus “animats” en una estructura plana de cromosoma. La mayoría de las representaciones son altamente flexibles y de longitud variable, requiriendo operadores genéticos complejos.

Debido a que la investigación en este campo todavía está en una etapa inicial, las técnicas evolutivas se utilizan a menudo como herramientas de exploración, de una manera similar al Arte Evolutivo. Estos algoritmos se pueden utilizar para generar soluciones múltiples, incorporando competición, cooperación y otros métodos avanzados [Cliff, 94]. Sims, por ejemplo, coevolucionó a dos criaturas que competían entre sí simultáneamente para capturar un bloque virtual y demostró cómo la evolución puede desarrollar muchos diseños y contra-diseños. En este experimento una de las criaturas desarrolló un brazo de gran longitud para hacerse con el bloque rápidamente y la otra criatura desarrolló un brazo para bloquear este movimiento [Sims, 94].

La evaluación consiste generalmente en analizar el comportamiento en mundos virtuales simulados (aunque algunos investigadores prueban soluciones usando los robots reales [Harvey, 97]) y suelen necesitar de alta capacidad de cálculo. Para intentar acortar tiempos de ejecución de la evolución, se utilizan métodos avanzados, como por ejemplo AGs de estado estacionario, AGs paralelos, AGs híbridos [Cliff, 94]. Muchos sistemas que desarrollan este tipo de Diseño Evolutivo utilizan funciones de calidad variables, que hacen necesario el uso de otras técnicas genéticas de búsqueda especializadas [Harvey, 97]. La mayoría de los sistemas desarrollan formas desde cero (la población inicial es al azar), no obstante algunos generan de vez en cuando poblaciones iniciales con los individuos más aptos de las ejecuciones anteriores [Sims, 94].

El trabajo típico en Vida Artificial consiste en la creación de sistemas de control (o “cerebros”) para robots, capaces de producir comportamientos tales como selección de la acción, planificación y aprendizaje [Guillot, 94]. Por ejemplo, Harvey [Harvey, 94] ha desarrollado con éxito los “cerebros” y la morfología visual de robots capaces de

distinguir visualmente entre un rectángulo y un triángulo. Sin embargo, quizás el trabajo más notable de esta área es el de Karl Sims [Sims, 94], que desarrolla los “cerebros” y “cuerpos” de criaturas virtuales usando bloques rectangulares definidos jerárquicamente para definir formas simples de criaturas y un Algoritmo Genético para evolucionarlos. Sims ha desarrollado criaturas tipo tortuga con capacidades asombrosas como “nadar” y “seguir luces”, así como criaturas que “caminaban” y criaturas que “saltaban” [Sims, 94]. Los “animats”, tales como los desarrollados por Sims, demuestran claramente que el ordenador es capaz de desarrollar diseños funcionales desde cero sin ninguna interacción humana.

4.2 Diseño Evolutivo híbrido

Muchos investigadores se especializan en uno de los cuatro aspectos principales de Diseño Evolutivo sin plantearse la posibilidad de considerar aproximaciones alternativas. Sin embargo, más recientemente, algunos han empezado a combinar las ideas de estas áreas en su trabajo. Esto ha llevado a la creación de cuatro nuevos aspectos.

4.2.1 Diseño Evolutivo Integral

La evolución de diseños de ingeniería está extendiéndose debido a la proliferación de numerosos centros de investigación explorando este tipo de conceptos. Aunque la mayor parte de la investigación parece centrarse en la categoría de Optimización Evolutiva, o en la categoría de Diseño Evolutivo Creativo, algún trabajo intenta combinar ambos aspectos en lo que se llama sistemas de Diseño Evolutivo Integral.

Parmee [Parmee, 96] sugiere que el ordenador puede usarse en todo el proceso de diseño, es decir, tanto en las fases iniciales conceptuales de ingeniería básica como en las fases de ingeniería de detalle y usando en ambos casos técnicas adaptativas. Parmee habla de cómo el uso de varios sistemas diferentes, cada uno dedicado a una fase específica de diseño podría usarse en combinación, “integrando la búsqueda adaptativa en cada fase del proceso de diseño”.

Por contra, Bentley [Bentley, 96, 97] ha investigado el uso de un solo sistema de Diseño Evolutivo Genérico para realizar el proceso de diseño completo sin hacer distinción entre sus fases y con el objetivo de realizar múltiples tareas diferentes. El trabajo de Bentley ha demostrado que es posible usar el ordenador para evolucionar completamente nuevos diseños desde el principio, y optimizarlos, para que cumplan su criterio funcional específico

4.2.2 Diseño Evolutivo Creativo basado en la Vida Artificial

Los trabajos de investigación realizados en Vida Artificial han generado formas de asombrosa diversidad y creatividad, de tal modo que algunos investigadores están usando actualmente algunas de estas técnicas en su sistema de Diseño Evolutivo Creativo, en un esfuerzo por mejorar la calidad y originalidad de sus diseños técnicos evolucionados.

Por ejemplo, Parmee [Parmee, 96] en su método de “colonia de hormigas” usa métodos de agentes distribuidos de Vida Artificial para mejorar su búsqueda evolutiva y Coates [Coates, 97] emplea Programación Genética para evolucionar nuevas formas arquitectónicas.

Existen todavía muchas otras posibilidades inexploradas en esta área. Por ejemplo, Bonabeau en [Bonabeau, 94] describe a una forma de vida artificial generada por simulación que consiste en un enjambre de avispas artificiales que construyen arquitecturas tridimensionales tipo panel. Una línea futura de investigación podría ser la evolución de avispas artificiales capaces de construir nuevos diseños.

4.2.3 Vida Artificial Evolutiva Estética.

Los primeros trabajos de investigación sobre la evolución de Formas de Vida Estéticas fueron realizados por Dawkins [Dawkins, 86] que seleccionó manualmente sus “biomorphs” artificiales para la reproducción, del mismo modo que los artistas seleccionan sus formas usando sistema de Arte Evolutivo. Ventrella [Ventrella, 94] da un paso más y evoluciona sus “animats” con aspecto de palos animados guiados por el juicio estético del usuario y por su habilidad de caminar de forma natural en un mundo virtual. Alternativamente, Lund [Lund, 95] describe el uso de redes neuronales que aprenden a juzgar cuadros evolucionados basado en su estética.

Aunque ha habido pocas aplicaciones hasta la fecha que se beneficien esta área de investigación, el uso del ordenador para desarrollar caracteres animados atractivos o divertidos son de gran ayuda en la industria de los videojuegos y la televisión.

4.2.4 Diseño Evolutivo Estético

La evolución de los diseños estéticos es un área obvia de investigación que debe recibir más atención quizás de lo que se hace. Algunos diseños escogidos debido a su estética no son completamente funcionales y algunos funcionalmente excelentes son desechados debido a su mala apariencia. Furuta [Furuta, 95] describe una aproximación a este concepto en el que diseños de puentes pueden perfeccionarse morfológicamente usando AGs, por medio de “psicovectores” que cuantifican los factores estéticos de las estructuras.

En síntesis, los ordenadores de hoy en día son capaces de generar nuevos diseños y optimizar su desempeño mejor que el diseñador humano más experto. Tanto es así que el “diseño asistido por ordenador” se está convirtiendo en “diseño por ordenador”. Los sistemas de Diseño Evolutivo son herramientas software que ayudan al diseñador en el proceso de diseño para mejorar la productividad, calidad, velocidad y costos de un sistema gracias a su potencia de cálculo y bajo costo. Sin embargo, aunque se están obteniendo resultados impresionantes no dejan de ser herramientas diseñadas exclusivamente para determinados diseños con una flexibilidad muy limitada.

-
-
-

Entorno Evolutivo de Diseño Automático en Ingeniería

5 Entorno Evolutivo de Diseño Automático

El interés que suscita el estudio del proceso de diseño es tal que en el documento que la NFS "Research Opportunities in Engineering Design" publica en abril de 1996 se establece que dentro de las 12 líneas principales de investigación hasta el año 2006 las 4 de mayor impacto serán:

- Desarrollo de herramientas y técnicas de ayuda en el diseño colaborativo;
- Empleo de teorías normativas, métodos de diseño y modelos prescriptivos;
- Desarrollo de herramientas e infraestructura para la integración de las etapas de diseño; y
- Desarrollo de sistemas para el soporte a la información en el diseño.

La mayoría de las investigaciones realizadas en esta línea, encaminadas a mejorar las distintas etapas del proceso de diseño, adoptan la hipótesis de que la fase crucial es la de diseño conceptual en la que se generan las ideas y que se caracteriza por una gran incertidumbre debido a que el problema en ocasiones no está bien definido. Otras aproximaciones consideran que para diseñar de forma eficiente, además de mejorar la generación de soluciones, es importante controlar cómo conseguir la convergencia del problema. Como ya se ha indicado en secciones anteriores, estas etapas se dejan en general en manos del diseñador humano como consecuencia de la incertidumbre asociada a las mismas y a las reticencias en dejar el proceso de decisión en manos de un sistema automático.

Debido a que el entorno computacional de diseño que se presenta en esta tesis pretende desligar al diseñador del ciclo de diseño en la medida de lo posible se hace hincapié en las dos etapas fundamentales del diseño conceptual: la etapa de búsqueda o etapa divergente donde se generan los conceptos y la etapa de toma de decisiones o etapa convergente en la que se realiza la evaluación y selección de los mismos [Pugh, 90], [Cross, 94], [Roozenburg, 95].

En este capítulo se desarrolla este concepto partiendo de un breve resumen de las etapas de diseño analizadas en los capítulos anteriores, proporcionando una descripción de sus características principales y reformulando el problema de diseño en términos de un proceso de búsqueda en un espacio de soluciones normalmente de alta dimensionalidad y, dado que lo que se pretende es proporcionar caminos para la obtención de soluciones originales, de dimensión variable. Replanteado así el problema, se estudia cada uno de los procesos implicados en la obtención de diseños que cumplan las especificaciones planteadas y se determinan las opciones computacionales para llevarlos a cabo de una forma eficiente, estableciendo las estrategias y herramientas necesarias. Finalmente, y con el objeto de proporcionar una visión global y explicar la elección de los ejemplos planteados en los siguientes capítulos, se realiza una breve descripción de los mismos.

5.1 Punto de partida: Etapas del proceso de diseño

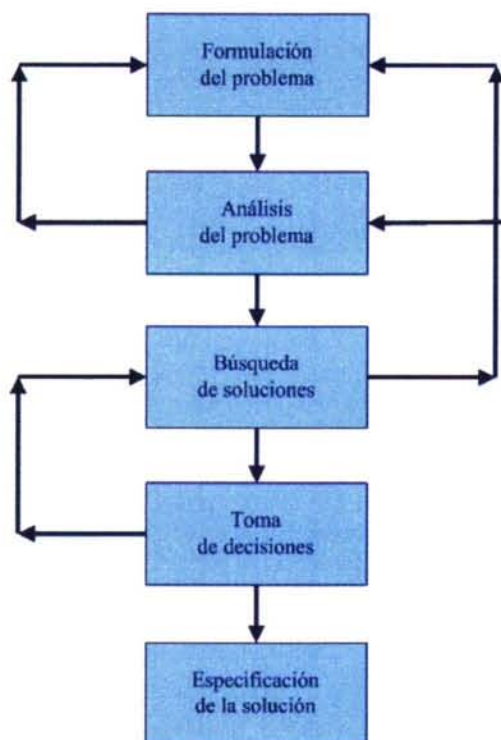


Figura 5.1: Etapas del proceso de diseño

Como ya hemos visto, el objetivo fundamental de la ingeniería es obtener el medio más adecuado para lograr la transformación de cierto estado de cosas en otro, o lo que es lo mismo, resolver un problema. Es difícil concebir un problema para el que sólo haya una solución, es más, para la mayoría de ellos existen más soluciones posibles de las que se pueden estudiar. Es por eso que todo procedimiento que ayude en la especificación de una solución satisfactoria a un problema determinado se hace imprescindible. Como ya se ha indicado anteriormente, este conjunto de actividades que sirve de apoyo al diseñador para la consecución de sus objetivos se denomina proceso de diseño y se podría dividir en cinco etapas fundamentales en función de las actividades a realizar:

1. Formulación del problema.

Consiste en la determinación del significado básico del problema identificando la transformación que se desea realizar respecto a los estados inicial y final. Fundamentalmente se busca determinar QUÉ queremos obtener.

Una formulación amplia maximiza el número y la variedad de las soluciones alternativas que pueden ser consideradas aumentando la probabilidad de obtener una solución general de gran calidad que pueda diferir en gran medida de las tradicionalmente adoptadas. En contraposición, la formulación estrecha de un problema o su subdivisión excluye enormes conjuntos de soluciones parciales potencialmente valiosas que dan lugar a otras generales menos efectivas.

2. Análisis del problema.

Esta etapa del proceso de diseño consiste en la definición al detalle de un problema traduciendo su descripción general en características específicas. El análisis hace hincapié en la definición y estudio de las variables de entrada y salida y de sus limitaciones, de las restricciones o características que toda solución parcial debe tener, de los criterios de selección de soluciones y de su ponderación.

A las primeras dos etapas del proceso de diseño genérico se las suele agrupar bajo el término de “Especificación del problema”. Su conclusión suele ser un documento en el que queda definido con claridad y precisión lo que se desea obtener y como se va a medir el cumplimiento de dicho objetivo.

3. Búsqueda de soluciones.

Una vez especificado el problema delimitando su alcance y lo que se pretende, las siguientes etapas tratan de resolver CÓMO obtenerlo por medio de procesos de búsqueda de soluciones y toma de decisiones de los caminos a seguir en esta búsqueda. Esta es la fase habitualmente designada como “diseño”.

La ingeniería es básicamente una actividad creadora que consiste en la búsqueda de soluciones alternativas para la resolución de problemas. Sin embargo, en contra de este precepto, en la mayoría de los procesos de diseño reales, las soluciones tienden a agruparse en torno a las tradicionalmente adoptadas basadas en el conocimiento humano y en la experiencia. Tanto es así que gran parte de los procesos de diseño realizados en ingeniería se basan únicamente en la consulta de la documentación existente.

Sin embargo, existe otro tipo de búsqueda de mucha más complejidad, no tan sencilla y controlable y propensa a la ineficiencia cuyo objetivo es el desarrollo de la inventiva. La capacidad creativa aumenta conforme se eliminan restricciones ficticias, se amplía el número y variedad de las soluciones y se realiza una búsqueda eficaz. Estos elementos y la forma de implementarlos habrán de ser uno de los hilos que guía el trabajo planteado en esta memoria.

4. Toma de decisiones.

Así como en la fase de búsqueda de soluciones aumenta el número y la variedad de las posibles soluciones, es necesario incorporar al proceso de diseño un procedimiento de eliminación que reduzca estas alternativas hasta obtener la solución definitiva.

Las etapas a seguir en esta fase se fundamentan en tres puntos: la selección de criterios y determinación de su ponderación relativa que son resultado de la etapa de especificación (en el proceso de diseño tradicional, los diseñadores traducen estos criterios de especificación en el dominio del problema a otros criterios basados en parámetros del dominio de implementación), la predicción del funcionamiento de las soluciones alternativas mediante el uso de modelos matemáticos, prototipos o simulación, su comparación en función de los criterios y, por último, la elección del camino a seguir o la solución adoptada. Es importante volver a resaltar la dicotomía existente en el proceso de diseño tradicional entre los criterios indicados en el

documento de especificación del problema, normalmente desarrollado de acuerdo con los criterios del cliente al que va dirigida la solución y, por lo tanto, en su dominio (rendimiento económico, capacidad de producción, etc.) y los criterios del diseñador en el dominio de la implementación, relacionados con parámetros de la tecnología utilizada para desarrollar las soluciones (consumos eléctricos, parámetros de control, etc.)

Respecto a los criterios en el dominio del problema es necesario tener en cuenta la imposibilidad o excesiva dificultad de la cuantificación de muchos de ellos (seguridad, facilidad de manejo u operabilidad, aceptación del público,...) y la dificultad que entraña la ponderación cuando existen conflictos. El criterio que presenta más dificultades es la calidad estética pues es función de la percepción del observador y requiere que el diseñador anticipe gran variedad de respuestas individuales en muchos casos difícilmente traducibles a criterios en el dominio de implementación.

5. Especificación de la solución adoptada.

En esta etapa se realiza la especificación en detalle de las características de la solución alternativa seleccionada en la fase de decisión mediante la elaboración de planos de detalle, informes técnicos o modelos para prueba y demostración. Una especificación adecuada es fundamental en el ciclo de diseño para la implantación correcta de las soluciones en fabricación y servicio. Aún así, la supervisión periódica de la instalación y operación de las soluciones adoptadas es fundamental para mejorar diseños futuros o rediseñar.

A partir del análisis realizado sobre el proceso de diseño y sus etapas se observa que, en resumen, un buen diseño es el que haciendo uso óptimo de los recursos disponibles es capaz de encontrar soluciones a un determinado problema y cumplir con las especificaciones definidas tan bien como sea posible. Teniendo en cuenta esto, diseñar es básicamente un proceso de búsqueda y, como tal, son aplicables las técnicas que se usan en cualquier otro campo. En el caso de diseños complejos donde el espacio de búsqueda llega a ser enorme y se hace imposible explorar eficientemente todas las opciones disponibles, las estrategias tradicionales de búsqueda no pueden alcanzar soluciones adecuadas en un tiempo razonable pues se basan en la percepción, experiencia e intuición del diseñador dejando a un lado la exploración exhaustiva.

Para hacer frente a este problema sería deseable eliminar al diseñador humano del bucle de diseño o, al menos, minimizar su participación a las dos etapas iniciales del proceso de diseño, es decir, a la formulación y análisis del problema, esto es, a la especificación del mismo. Esta necesidad de liberar al diseñador humano de su función clásica de búsqueda y toma de decisiones en la resolución de sistemas de gran complejidad es el punto de partida para el entorno de diseño desarrollado y busca, por una parte, mejorar la creatividad, eliminando restricciones ficticias introducidas por el diseñador humano para hacer los problemas manejables, y, por otra, mejorar las soluciones al trabajar directamente sobre parámetros en el dominio del problema sin operar sobre la traducción ya realizada por un humano con las restricciones que ello conlleva.

Sustituir al humano como diseñador y relegarlo a una función de especificación del problema necesita de un entorno automático que, a partir de las especificaciones y recursos de que se disponen, controle el proceso de búsqueda de posibles soluciones en función de la experiencia previa y la creatividad. Este proceso de búsqueda no consiste

en encontrar una única solución a un problema, sino en seleccionar de un abanico de alternativas una solución definitiva estimulando constantemente la innovación y la integración de todos los factores (funcionales, culturales y de producción) que intervienen en el proceso.

La necesidad de buscar una amplia variedad de soluciones parciales obliga a realizar un gran esfuerzo de cálculo y a sistematizar el proceso de síntesis, combinación y evaluación de las mismas evitando al mismo tiempo la aparición de restricciones ficticias que reduzcan la zona de exploración. Además si el criterio de búsqueda una vez realizada la evaluación de cada una de las soluciones consiste en dar preferencia a unas en relación a otras, el proceso se simplificaría en gran medida. Por este motivo la automatización del proceso no asegura su éxito pues a menudo se hace inviable explorar todo el espacio de soluciones o hacer uso de una búsqueda aleatoria. Se hace necesario, por lo tanto, la incorporación de técnicas avanzadas de búsqueda.

Con respecto a la etapa de toma de decisiones debe tenerse en cuenta que en el caso de diseños de alta complejidad es necesario evaluar la calidad de cada una de las múltiples soluciones parciales generadas por la etapa anterior y que en ocasiones, al no ser viable económicamente la evaluación de un sistema bajo condiciones reales de funcionamiento, se recurre al uso de entornos de simulación. Estas herramientas computacionales de análisis son cada vez más avanzadas, ahorran tiempo y dinero y facilitan la automatización del proceso en gran medida. Aún así, cuando en esta etapa se requiere de la participación de uno o mas usuarios sería interesante hacerlos partícipes de la evaluación de las soluciones obtenidas en función de su percepción personal dotando al entorno computacional automático de diseño de cierta interactividad.

Es de destacar que aunque son muchos los avances realizados en el desarrollo de metodologías computacionales para la optimización del diseño, rara vez lo hacen de forma conjunta y las que lo consiguen son específicas para el desarrollo de determinados productos careciendo de versatilidad. Dicho de otro modo, no existe actualmente una metodología que guíe al diseñador en el proceso de diseño siendo un reto llegar a alcanzar un sistema computacional completo de soporte al diseño que asista al diseñador en su cometido. Es importante no confundir guiar con planificar ya que esta última es una tarea previa al diseño mientras que el guiado actúa durante el mismo.

En síntesis, la automatización del proceso de diseño de sistemas y productos que propone la metodología expuesta en esta tesis se basa en la transformación del problema de diseño en un problema de búsqueda en un espacio de soluciones de dimensionalidad en general alta y no siempre conocida a priori y, por lo tanto se apoya en herramientas computacionales de búsqueda y evaluación. Las primeras están representadas por técnicas evolutivas debido a sus características de flexibilidad y versatilidad y las segundas por simuladores avanzados de alta fiabilidad que se ejecutan en un entorno distribuido escalable para aumentar su potencia y disminuir su tiempo de cálculo. Estos simuladores integrados en el entorno pueden realizar la evaluación técnica y automática de cada una de las soluciones parciales sintetizadas o visualizarlas y permitir la opinión subjetiva de expertos o usuarios finales mediante una interfaz hombre-máquina. La incorporación de parámetros subjetivos confiere al entorno de cierto carácter concurrente.

5.2 Características deseadas de un entorno de diseño

A modo de resumen se enumeran a continuación las características que un entorno de diseño automático debería tener.

- Búsqueda y análisis computacional de soluciones alternativas

El objetivo principal de todo entorno computacional de diseño es auxiliar al diseñador en su cometido tomando como referencia el proceso de diseño básico, imitando así las fases por las que en teoría se sintetiza la solución a un determinado problema. En este sentido el entorno debe ser capaz de implementar las fases de búsqueda y toma de decisiones para agilizar el proceso de diseño y minimizar los medios necesarios.

La implementación en el entorno de estas dos etapas hace imprescindible la sustitución de la búsqueda tradicional por otra más avanzada y la experimentación real de un diseño por simuladores especializados actuales cada vez más precisos, rápidos e integrables. Un sistema de este tipo con una metodología adecuada no estaría limitado al desarrollo de soluciones cercanas a las ya existentes y potenciaría la síntesis de soluciones innovadoras radicalmente distintas a las creadas por un diseñador humano. Del mismo modo podría constituir un nuevo entorno de diseño muy útil a la hora de sintetizar soluciones a problemas de excesiva complejidad como para ser abordadas por otro tipo de técnicas más tradicionales. Resumiendo, se desearía que el entorno pudiese emular el proceso de diseño que seguiría un humano en cuanto a exploración de un espacio de posibles soluciones y evaluación de las mismas.

- Automatización del Proceso de Diseño

La búsqueda de diseños alternativos e innovadores necesita que el espacio de búsqueda de soluciones sea lo más amplio posible en un intento de aumentar la variedad de las posibles soluciones candidatas. Al mismo tiempo sería deseable que el entorno computacional de diseño fuese totalmente transparente al diseñador pues así no intervendría en tareas que necesitasen de conocimientos específicos como, por ejemplo, el manejo de simuladores dejando sólo a su control las fases de formulación, análisis y configuración básica del sistema.

Todos estos requerimientos se hacen viables en el caso de entornos computacionales automáticos de diseño que requieren de la integración de las herramientas computacionales que lo conforman para su colaboración. Es de destacar que aunque algunos tipos de software facilitan en gran medida este objetivo (algunos permiten un alto nivel de personalización mediante el uso de lenguajes de programación) son muchos los que no están concebidos para este propósito provocando que la integración se haga en ocasiones complicada y requiera de un gran esfuerzo.

- Metodología de búsqueda avanzada

Por muy automático que sea el proceso de diseño, la búsqueda de soluciones no puede limitarse a la exploración completa ni aleatoria de todo su espacio pues sería inviable en el caso de problemas de relativa complejidad. Es por esto que se hace necesaria la participación en el entorno automático de diseño de técnicas de búsqueda avanzadas.

- Conveniencia de contemplar computación distribuida

Con todo, aunque el entorno automático de diseño presente una metodología avanzada de búsqueda de soluciones y las herramientas informáticas sean cada vez más rápidas, la experiencia nos dice que esto llevará a un aumento cada vez mayor de la complejidad de los problemas a tratar y, por lo tanto, de su coste computacional. Por lo tanto, si se pretende realizar un sistema escalable de una forma suave, se debe diseñar de manera que también sea escalable en cuanto a los recursos computacionales que necesite. Consecuentemente, se plantea la necesidad de que su implementación contemple una ejecución distribuida con distintos niveles de granularidad según se requiera para cada problema concreto.

- Búsqueda de facilidad de interacción con el usuario

La interacción múltiple humana de diseñadores y/o usuarios será factible cuando el proceso de diseño dependa de criterios subjetivos. Esta percepción personal puede conformar total o parcialmente la evaluación de una determinada solución. En este último caso la intervención humana se complementa con otros criterios objetivos provenientes de herramientas computacionales de evaluación. Esta participación humana en la fase de toma de decisiones no debe confundirse con la tarea ineludible del diseñador en las fases iniciales de formulación y análisis del problema. Por lo tanto, sería deseable disponer de una metodología para la interacción hombre-máquina de manera que un usuario o conjuntos de usuarios del sistema pudiesen interactuar con él de manera que sus "opiniones" pudiesen ser tenidas en cuenta sin eliminar el carácter de sistema de diseño automático que gobierna este esfuerzo. De hecho, el permitir este tipo de interacción nos lleva a una posible aproximación para la incorporación de técnicas de diseño concurrente a entornos de este tipo.

- Necesidad de un sistema modular

La división del proceso de diseño en cinco etapas fundamentales y la búsqueda de un entorno de diseño automático versátil capaz de desarrollar una amplia gama de diseños con la mínima reconfiguración da a entender que la estructura modular del entorno puede aportar ciertas ventajas.

La estructura modular del sistema posibilita la obtención de una variedad de diseños relativamente alta mediante la intercolaboración de componentes básicos de una funcionalidad concreta directamente relacionada con las etapas del proceso de diseño que se encapsulan de forma estándar con un sistema de interfaces bien definidos a tal efecto. Siendo realistas, sin embargo, parece probable que ciertos tipos de problemas causarán dificultades para el sistema en cuyo caso será implementada la reconfiguración necesaria. Este tipo de reconfiguración consistirá, por ejemplo, en adaptar el software de evaluación al sistema y/o modificar parámetros en la metodología de búsqueda (operadores, probabilidades de aplicación, codificación, calidad, etc.) en función de las especificaciones definidas por el usuario. Por lo tanto, se ha dedicado en este trabajo un gran esfuerzo a tratar de diferenciar los módulos que componen el sistema de diseño global así como a definir sus interfaces de manera que el entorno fuese razonablemente fácil de adaptar a nuevas propuestas.

5.3 Bloques constructivos básicos

El objetivo de la modularidad del entorno de diseño es la combinación de diferentes herramientas que intercolaboran entre sí para obtener diseños complejos que trabajan en el mundo real en poco tiempo y a costo muy bajo. La versatilidad de esta estructura permitirá adaptar rápidamente el sistema para el diseño de un determinado producto.

Los bloques principales de este sistema de carácter modular son:

- Módulo de búsqueda de soluciones.
- Módulo de toma de decisiones (evaluación con o sin interacción humana).
- Módulo de distribución computacional.

En el siguiente gráfico se muestra el esquema funcional del Entorno Evolutivo de Diseño Automático desarrollado.

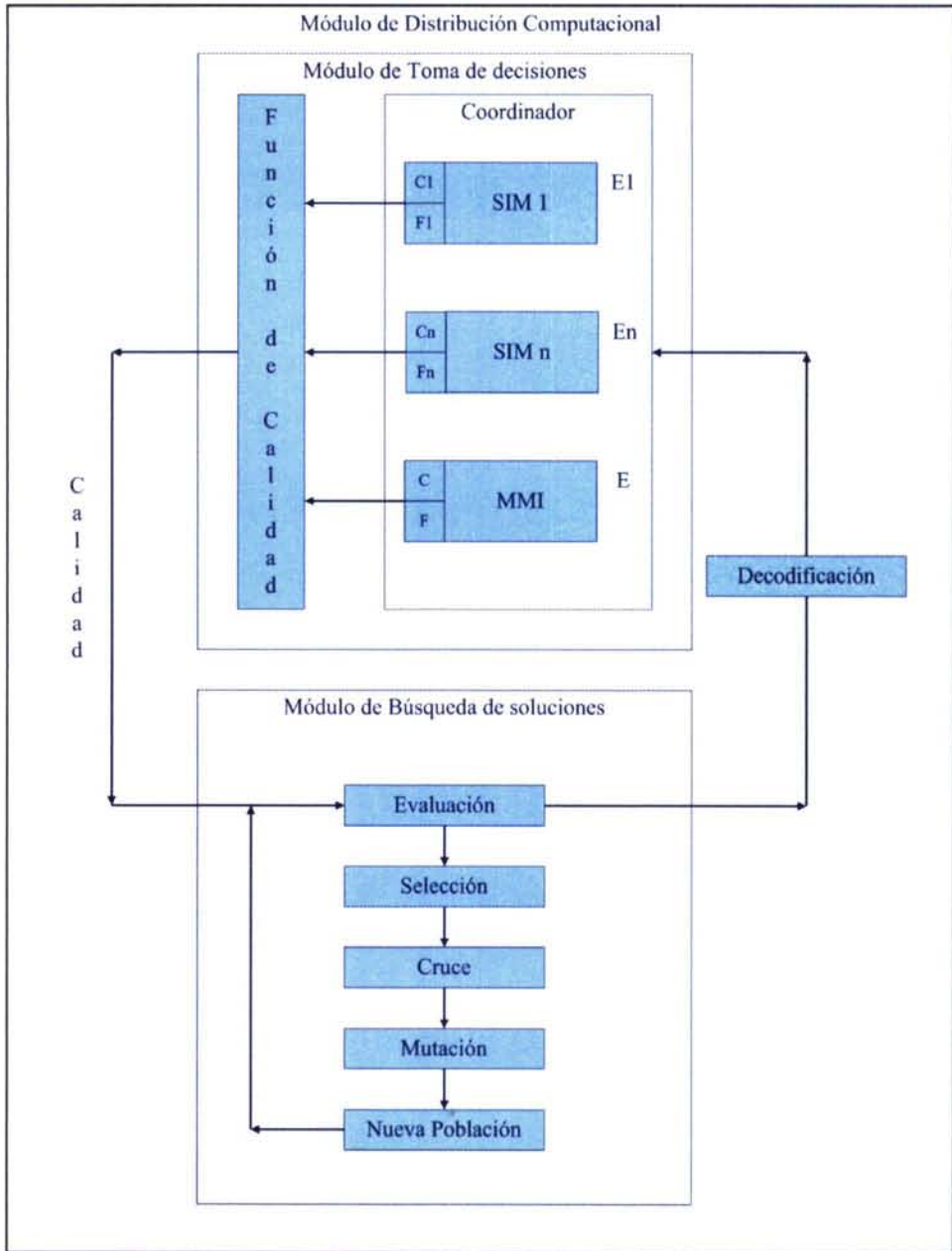


Figura 5.2: Esquema del Entorno Evolutivo de Diseño Automático.

El módulo de Toma de decisiones está formado por un conjunto de simuladores e interfaces hombre-máquina que se integran entre sí con la ayuda de un coordinador. Las salidas de cada uno de los simuladores son entradas de una función de calidad que también forma parte del módulo.

La salida de la función de calidad es un valor o vector de aptitud del que hace uso el bloque de evaluación del módulo de Búsqueda de Soluciones. Los operadores genéticos permiten obtener una población de soluciones parciales de mayor calidad en función de la aptitud de cada uno de sus individuos.

Una vez obtenida una nueva generación de soluciones, es necesaria una decodificación que permite traducir el lenguaje propio del módulo de búsqueda (genotipo) al de toma de decisiones (fenotipo).

En el módulo de Toma de Decisiones (simuladores, interfaz hombre-máquina o MMI y función de calidad) destacan los siguientes elementos:

- Encapsulador del simulador (En):

Implementa una interfaz estándar entre el simulador y el entorno que incluye:

1. Decodificación del fenotipo a los parámetros del modelo de simulación (pesos de una red neuronal, número y disposición de máquinas, conexionado de elementos electrónicos, superficies en el espacio tridimensional, etc.).
2. Gestión de los parámetros de la simulación (tiempo de simulación, número de simulaciones, etc.) a través de scripts o de la extracción de características del propio cromosoma.
3. Gestión de las órdenes recibidas por el coordinador de procesos.
4. Salidas:
 - a. Órdenes o comandos destinados a otros procesos o simuladores que gestiona el coordinador de procesos.
 - b. Valores de la evaluación del simulador que forman parte de la función de calidad y cuyo resultado es tratado en el evaluador del módulo de Búsqueda de Soluciones.

- Filtro de parámetros (Fn):

Módulo que selecciona qué parámetros se tienen en cuenta de todos los que puede generar un simulador para formar parte de la función de calidad o ser tratados como señales de control para otros procesos.

- Señales de control (Cn):

Señales que coordinan la colaboración entre las distintas herramientas de simulación.

A continuación pasamos a describir más en detalle cada uno de los elementos que participan en esta estructura y sus relaciones. Posteriormente se definirán cada uno de los bloques del entorno para los distintos experimentos realizados con la intención de personalizar el sistema y aclarar sus conceptos.

5.3.1 Módulo de búsqueda de soluciones

A la hora de la implementación del módulo de búsqueda de soluciones es necesario tener en cuenta el tipo de representación que se utiliza y, por lo tanto, el espacio de búsqueda que determina. Como ya se ha indicado, la aproximación desarrollada en esta tesis implica la transformación del proceso de diseño en un proceso de búsqueda dentro de un espacio de posibles soluciones. Este espacio viene caracterizado por una parametrización del problema a veces no conocida a priori. Si además pretendemos que el proceso de diseño sea creativo, la dimensionalidad del espacio de búsqueda puede cambiar a lo largo del proceso de exploración debido a la existencia de una reordenación de las soluciones en función de su aptitud. Este proceso genera una topología casi siempre desconocida a priori y que es necesario explorar para encontrar soluciones óptimas evitando estancarse en soluciones que no lo son. Por este motivo es evidente que el algoritmo de exploración a emplear debe proporcionar unos rendimientos muy altos.

De las distintas alternativas barajadas para la implementación del proceso de búsqueda se ha optado por las técnicas evolutivas por las razones que a continuación se indican:

- Minimizan la posibilidad de quedarse en extremos locales
- Proporcionan familias de soluciones (no una única). Esto es especialmente interesante en el campo del diseño ya que a menudo existen requisitos contradictorios, o de compromiso. Lo que se pretende, en definitiva, es la obtención de un frente de Pareto de soluciones.
- Utilizan operadores que permiten la generación de soluciones parciales.
- No requieren nada más que una función objetivo para su funcionamiento, lo que permite incluso trabajar en espacios con discontinuidades.
- Su implementación es fácilmente paralelizable y distribuable.
- Permiten su combinación de manera eficiente con otros tipos de algoritmos allí donde resulte necesario.
- El proceso que siguen, como ya se ha indicado, es paralelo al proceso de diseño del humano.

En definitiva, los algoritmos evolutivos son algoritmos de búsqueda adaptativa basados en la supervivencia del más apto cuya flexibilidad, eficiencia, robustez, similitud conceptual con el diseño humano y facilidad de paralelización hace que sea el algoritmo seleccionado para implementar el módulo que nos ocupa.

En todo algoritmo evolutivo se parte inicialmente de una población aleatoria de individuos donde cada uno de ellos representa una posible solución codificada como un cromosoma. El conjunto de parámetros codificados de un individuo se conoce como genotipo, mientras que la solución que definen estos se conoce como el fenotipo.

Una vez seleccionados los individuos de la población con una probabilidad proporcional a su calidad se genera un conjunto de reproducción en donde los cromosomas son sometidos a un proceso de evolución y son manipulados por una serie de operadores genéticos (cruce y mutación) en función de la evaluación del individuo que representan. Este proceso de evaluación se realiza según una función de calidad previamente normalizada para posteriormente sintetizar nuevos individuos que reemplazarán a sus predecesores.

En el entorno que aquí se describe y por razones de generalidad y simplicidad operativa se ha optado por la implementación de un Algoritmo Genético Simple como motor de la búsqueda evolutiva planteada. Este algoritmo, como cualquier genético, requiere para su implementación computacional de una codificación del espacio de soluciones, un mecanismo de selección, operadores genéticos de cruce y mutación y una función de calidad o aptitud. En los puntos que siguen se comentan en profundidad estos aspectos.

- Codificación

Se pretende que el entorno de diseño esté abierto a la inclusión de múltiples tipos de codificación en función de las necesidades de cada problema. Estas codificaciones pueden ser directas, donde genotipo y fenotipo son idénticos, o indirectas cuando a partir del genotipo el fenotipo se obtiene a partir de un proceso de construcción o desarrollo. Lo que se ha buscado, en general, es aislar al algoritmo evolutivo de la codificación de manera que trabaje, siempre que sea posible, con cadenas de números reales, independientemente de lo que éstas representen y que se decodifican en un módulo de interfaz externo donde se adaptan al resto del sistema. Aún así, y para añadir versatilidad, a cambio de mayor manipulación del entorno, se ha preservado la posibilidad de que el evolutivo trabajase con otros tipos de representación por medio de su estructura interna modular donde los distintos operadores pueden ser sustituidos en función de las necesidades.

En los ejemplos que se presentan en los siguientes capítulos se proponen ambas aproximaciones. Por una parte, en el caso de los controladores diseñados para la maniobra y estabilidad de submarinos se hace uso de fenotipos definidos mediante redes neuronales en donde los individuos de la población son codificados como pesos de la red. Estas redes neuronales equivalen a aproximadores universales que permiten una determinada actuación sobre del vehículo en función de señales captadas por sus sensores. Por otra parte, cuando se consideró el diseño de circuitos electrónicos con puertas lógicas y memorias para el control de una planta industrial se aplicó una compleja codificación con estructura de árbol binario en donde además fue necesario rediseñar los operadores genéticos del algoritmo. Finalmente, en el diseño morfológico de plantas industriales y formas 3D la codificación basada en una secuencia de parámetros resultó ser más sencilla.

Es evidente, como en casi todos los campos de la ciencia, que la codificación ha de hacerse a medida del problema y lo importante en el desarrollo de éste módulo fue aislar la codificación de los procesos evolutivos implicados en el tratamiento de los genotipos.

- Operador selección

El operador que se ha utilizado para este módulo en todos los casos prácticos es el de selección por torneo por ser computacionalmente más eficiente, rápido y fácil de paralelizar. El criterio fundamental para la elección de este operador ha sido el hecho de que no requiere operaciones que impliquen la globalidad de la población, lo que resulta más fácil para su ejecución distribuida.

El mecanismo de un operador de selección por torneo implica seleccionar como padre al individuo más apto de entre dos o más de la población elegidos al azar en función de la presión selectiva que se quiera imponer. Evidentemente cuanto mayor

sea la presión selectiva, más rápido es el proceso convergencia pero mayor será la probabilidad de estancarse en extremos locales. Como norma general y dado el desconocimiento a priori de la complejidad de los espacios a contemplar, se ha utilizado una presión selectiva muy baja con torneos entre dos individuos.

- Operador cruce

El operador de cruce es muy fácil de modificar de acuerdo con el criterio del usuario, pudiendo recurrir a diversas implementaciones tales como cruce por un punto, por dos puntos, aleatorio y otros más específicos. Esta facilidad viene determinada por el aislamiento, en general, entre la codificación del problema y el evolutivo que trabaja directamente con cadenas de números reales. En la mayor parte de los casos hemos aplicado el operador cruce de dos puntos que reduce el efecto “bias posicional” y “punto final”. Este operador selecciona dos posiciones aleatoriamente para que entre ambas se intercambien los valores de los cromosomas. Sin embargo, como ya se ha indicado, se ha desarrollado un operador de cruce particular para la codificación en árbol binario de uno de los ejemplos tratados.

- Operador mutación

Consiste en modificar el valor de los genes de la totalidad de los cromosomas que definen la población en función de una determinada probabilidad. En el entorno desarrollado se permite casi cualquier tipo de operador de mutación, aunque en la mayor parte de los casos de aplicación que se presentan se aplica un operador tradicional aleatorio.

A pesar de sus grandes ventajas es necesario hacer notar que la aplicación de este tipo de algoritmos no asegura la convergencia de la población ni que la solución obtenida sea la óptima, pues ésta depende de la codificación aplicada, el tamaño de la población, el número de generaciones, la función de calidad, el tipo de operadores genéticos y la probabilidad aplicados. Sin embargo, es posible adaptar los parámetros que caracterizan al algoritmo en función de cada caso para asegurar el éxito del mismo. Por ejemplo si se pretende evitar la convergencia prematura de una población se puede incrementar su dimensión, mantener la diversidad de los cromosomas, controlar la presión selectiva, controlar la edad de los individuos, preservar los mejores, etc.

De todos modos el objetivo que se pretende con la implementación de este tipo de estrategias no es la optimización de un problema existente si no definir un entorno de diseño automático cuya metodología permita obtener un conjunto de posibles soluciones a gran variedad de problemas con la mínima reconfiguración posible.

La implementación global sobre el que se sustenta el entorno de diseño automático se divide en dos módulos, el módulo de búsqueda comentado en este apartado, que se realizó basándonos en las características de los algoritmos evolutivos, y el módulo de toma de decisiones del que hablaremos a continuación.

5.3.2 Módulo de toma de decisiones. Simulación/Interacción.

En términos de diseño el proceso de selección consiste en proporcionar al entorno la calidad de cada individuo candidato mediante su evaluación para que sea posible decidir un camino que merezca la pena seguir. Un algoritmo evolutivo requiere un valor para

esta calidad, bien sea un valor absoluto o un valor relativo en función del orden de las calidades de los individuos.

A partir de los resultados obtenidos en el módulo anterior, existen funciones que modelan la calidad de cada uno de los individuos de una población ponderando sus valores según una ecuación matemática. En esta función de calidad pueden intervenir aptitudes provenientes de uno o varios simuladores, comerciales o no, e incluso de la opinión subjetiva que un conjunto de usuarios tienen sobre cada individuo en particular.

En este sentido, en el entorno de diseño automático desarrollado se pretendió que la calidad de los diseños fuese evaluada según parámetros de calidad que podríamos denominar técnicos, objetivos o “de ingeniería” que aprovecharían al máximo la posibilidad de integrar al entorno simuladores comerciales o ad hoc que proporcionasen valoraciones del comportamiento de los diseños. Por otra parte, sería de gran utilidad que los parámetros de calidad pudieran ser también subjetivos incorporando la posibilidad de integrar en el entorno una interfaz que permitiese al usuario interactuar con él de forma natural.

Otro aspecto importante es que la función de calidad puede depender directamente del genotipo, como en el caso de la evolución morfológica que analizaremos más adelante, o del fenotipo generado mediante algún tipo de transformación. Este último caso es el más recurrido en los ejemplos defendidos en esta tesis y a menudo la función de calidad de un individuo depende de varios fenotipos al mismo tiempo.

La modularidad del entorno de diseño posibilita la integración fluida y simple con simuladores de todo tipo (mecánicos, hidrodinámicos, de control, etc.) comerciales o no a través de su adecuado encapsulamiento y el desarrollo de un módulo de coordinación de forma que cada uno de los individuos de una población puedan ser evaluados en condiciones lo más cercanas a la realidad posible. No cabe duda que el uso de software específico experto permite ahorrar un tiempo considerable a la hora de programar uno propio pero a veces o no existen o su generalidad de aplicación los hacen excesivamente lentos. Obviamente, la utilidad de los resultados obtenidos será tanto mejor cuanto más precisos sean los simuladores utilizados.

Los simuladores que implementan el módulo de toma de decisiones y que han sido utilizados en esta tesis son Matlab, Simulink, DADS, Extend y AutoCAD. Los dos primeros ambos de Mathworks y su toolbox de Control han sido utilizados en solitario y en colaboración con DADS (simulador dinámico de CADSI) para el control de maniobra y estabilización de robots submarinos respectivamente. Su versatilidad y amplia utilización en control es la causa de su incorporación en el entorno de diseño.

Simulink es una interfaz de programación de alto nivel gráfico y modular que permite la definición de las ecuaciones diferenciales que rigen el comportamiento de cualquier sistema y de la síntesis de sus controladores. A pesar de su extenso uso no está preparado para su ejecución automática y mucho menos paralela y ha tenido que ser modificado para tal fin. DADS, un simulador dinámico comercial permite exportar como una caja negra las ecuaciones de movimiento de cualquier sistema mecánico e integrarlas en Simulink. Sin embargo el proceso de generación de esta caja negra, esto es, su encapsulamiento, es manual y ha sido necesario programar su automatización. DADS se ha utilizado para modelar submarinos y robots hexápodos. Extend, por otro lado, es un simulador de procesos discretos muy utilizado para plantas industriales y que posibilita su programación en Visual Basic Automation (VBA). Éste es, por tanto,

el simulador más avanzado en cuanto a reconfiguración automática pero tampoco está preparado para su ejecución paralela en un cluster de ordenadores. Otra desventaja es que el control que se realice sobre la planta no puede implementarse en Matlab pues no posibilita esta integración con todas las limitaciones que esto conlleva. AutoCAD adolece de las mismas limitaciones que Extend pues permite su programación externa desde VBA pero no está preparado para su ejecución paralela.

Por lo tanto, la integración del módulo de toma de decisiones con el entorno de diseño es diferente en cada caso y supone en ocasiones un esfuerzo adicional. En el transcurso de la realización de esta tesis se ha observado que de forma paulatina los simuladores han avanzado hacia una integración cada vez mayor y muchos de ellos son programables con VBA (Extend y AutoCAD). De todas formas no lo están tanto a la hora de su ejecución automática y paralela pues están concebidos para ser utilizados por un único usuario y siempre a través de su interfaz gráfico.

La evaluación interactiva por medio de interfaces hombre-máquina también es integrable con el sistema y supone un importante complemento para este módulo. Este submódulo de evaluación interactiva dota al sistema principal de una interfaz hombre-máquina que posibilita la evaluación humana múltiple y remota. La interacción puede ser realizada por expertos que evalúen las soluciones alternativas que se van diseñando conforme se avanza en el proceso o usuarios finales del producto a diseñar.

En este último caso si el medio sobre el cual se sustenta la interfaz es Internet el sistema podría ser aplicado para la realización de estudios de mercado y proporcionar una herramienta sencilla y económica con la que sería posible diseñar productos que se adapten al máximo a los gustos y necesidades de un perfil determinado de usuarios. Así, teniendo en cuenta que el éxito de un producto en el mercado depende del equilibrio existente entre las especificaciones funcionales y el diseño, el sistema podría ayudar en gran medida a la consecución de dicho equilibrio.

Un equipo de ingenieros con experiencia en el sector, todo tipo de software para cálculo de propiedades térmicas, mecánicas,... y una muestra estadística que representase el perfil del público sobre el que va dirigido el producto, conformarían una marco función de calidad sobre la que se evaluarían posibles soluciones parciales inicialmente aleatorias que genéticamente evolucionarían hacia nuestro producto ideal. En este tipo de ingeniería concurrente la función de calidad estaría conformada por un conjunto de evaluaciones que irían desde la forma estética pura donde participaría directamente el usuario hasta la evaluación del modelo según el cumplimiento de especificaciones y normativas técnicas.

5.3.3 Módulo de distribución computacional

La aplicación en el módulo de búsqueda de algoritmos evolutivos a problemas con grandes poblaciones y espacios de exploración y las costosas evaluaciones de las soluciones parciales del módulo de toma de decisiones hace necesario la paralelización de ambos en un sistema de diseño distribuido que haga viable los tiempos de respuesta del entorno.

Esta demanda de capacidad de cómputo común en campos como el tratamiento de imágenes y gráficos, predicciones meteorológicas y entornos de simulación como el que nos ocupa, es consecuencia de la necesidad de realizar gran cantidad de cálculos y obtener resultados precisos en un tiempo razonable. Dado que la evolución tecnológica,

aún al ritmo vertiginoso al que avanza, no puede dar una respuesta satisfactoria a estos problemas, la solución pasa por los sistemas distribuidos y la programación paralela.

Un sistema distribuido está compuesto por recursos informáticos autónomos físicos y lógicos de propósito general que conectados mediante una red de comunicaciones y gracias a un sistema operativo comparten recursos hardware, software y datos para funcionar coordinada y transparentemente en la consecución de una determinada tarea que requiere de una elevada potencia de cálculo.

Por otro lado, la programación paralela es un concepto que data de los años 60 y hasta el inicio de los años 90 era un campo restringido por motivos económicos a centros de investigación y grandes empresas. Sin embargo, dos importantes novedades han hecho cambiar esta perspectiva: la aparición de las arquitecturas basadas en clusters con su excelente relación coste/beneficios y la introducción de MPI como estándar de programación para aplicaciones paralelas basadas en paso de mensajes.

La paralelización de algoritmos evolutivos no es una idea nueva. En la bibliografía podemos encontrar dos líneas de investigación en esta línea [Adamidis, 98] y [Cantú-Paz, 01] :

- Paralelización estándar o global: Un sólo procesador maestro supervisa toda la población y realiza la selección, los procesadores esclavos reciben los individuos para evaluarlos y, a veces, aplicarles los operadores genéticos.
- Paralelización por separación: Toda la población se encuentra distribuida bien como subpoblaciones o bien como individuos.

Por otra parte, podemos clasificar los algoritmos evolutivos paralelos según la forma de realizar sus comunicaciones:

- Comunicaciones síncronas: En este caso el algoritmo evolutivo para y espera para recibir información de otro procesador antes de proceder con la siguiente población [Collins, 91].
- Comunicaciones asíncronas: En este esquema el algoritmo evolutivo no se detiene para esperar a otros procesadores más lentos.

En nuestro caso y con idea de distribuir el entorno de diseño en una red de 50 ordenadores con características técnicas muy distintas se optó por paralelizar el módulo de búsqueda y toma de decisiones del sistema por medio de comunicaciones asíncronas y de modo estándar. Una vez definido el tipo de paralelización y comunicación se debe seleccionar la biblioteca de computación paralela sobre la que se sustentará el entorno distribuido.

Las principales bibliotecas de computación paralela son PVM (Parallel Virtual Machine) que se basa en el concepto de máquina virtual, esto es, una red de estaciones de trabajo manejados como un ordenador paralelo y MPI (Message Passing Interface) que se fundamenta en el paso de mensajes y se ha convertido en el nuevo estándar. Sin embargo, existen otras muchas alternativas como la utilización de bibliotecas de programación propias para ordenadores paralelos (NX en el Intel Paragon, MPL en el IBM SP2, etc.), lenguajes de programación paralelos (FORTRAN 90) o secuenciales (C, FORTRAN 77) con directivas de paralelismo o lenguajes secuenciales junto con compiladores que paralelicen automáticamente.

De todo este abanico de opciones se ha seleccionado la interfaz estándar MPI basada en paso de mensajes para paralelizar el entorno. La principal ventaja del MPI es que no exige una determinada implementación si no que el programador tiene a su disposición una librería de funciones para el diseño de aplicaciones sin necesitar del conocimiento del hardware sobre el que se va a ejecutar y la implementación de las funciones que emplea.

Otra ventaja fundamental es que es el método de programación de sistemas paralelos más potente y aceptado siendo considerado por algunos como el ensamblador de la programación paralela. Tanto es así que es soportado por la totalidad de los fabricantes de ordenadores paralelos y su portabilidad está asegurada a una gran variedad de sistemas: desde máquinas con memoria compartida hasta un "cluster" dedicado o una red local de ordenadores.

El modelo de programación que subyace tras MPI es SPMD (Single Program Multiple Data) un caso particular de MIMD (Multiple Instruction streams, Multiple Data streams) en el que cada uno de los procesos ejecutarán distintas sentencias del mismo programa en función de su rango. Estos procesos que tienen un espacio de memoria completamente separado se sincronizan mediante paso de mensajes.

Existen múltiples implementaciones específicas para sistemas concretos (Linux, MS Windows, IRIX, AIX, HP-UX) totalmente compatibles entre sí que permiten aprovechar al máximo las prestaciones de una máquina determinada. De entre todas ellas la versión más utilizada es mpich, desarrollada por el Laboratorio Nacional de Argonne y la Universidad de Chicago. Esta versión de libre distribución es la que se ha seleccionado para la paralelización del entorno de diseño desarrollado en esta tesis y ha sido exportada a numerosos sistemas operativos y máquinas, incluyendo clusters Linux y Windows.

Sin embargo, MPI aún está en sus comienzos, y aunque se está haciendo un hueco creciente en la comunidad de programadores de aplicaciones paralelas, todavía adolece de algunas limitaciones como que las comunicaciones deben programarse explícitamente, no existe un mecanismo estandarizado de E/S, no permite la creación dinámica de procesos, no existen bindings (concreciones de la interfaz) para otros lenguajes, además de C y FORTRAN, no es aplicable a sistemas en tiempo real y no existen interfaces gráficos de programación.

En nuestro caso en concreto se ha optado por la implementación de "mpich nt" la versión existente de MPI para entorno Windows NT programada en lenguaje C. Todos los experimentos que se abordarán en esta tesis se han implementado en un cluster bajo Windows NT de 50 PCs a 300 MHz.

Uno de los PCs, el servidor del cluster, se encarga de ejecutar el módulo de búsqueda del entorno de diseño y de distribuir mediante paso de mensajes todos y cada uno de los individuos que conformaban cada población. Así, en cada uno de los PCs restantes se ejecuta el módulo de toma de decisiones simulando el comportamiento de cada uno de los individuos. Una vez que se han simulado todos los individuos de la población el servidor del entorno ejecuta de nuevo el módulo de búsqueda para crear una nueva población y así hasta completar el objetivo.

El inconveniente más importante de esta implementación es el hecho de que al funcionar sobre mpich la asignación de máquinas es fija por lo que la pérdida de una máquina del cluster da lugar a la parada del proceso y la adición de alguna de ellas no

es automática. Sin embargo, en futuras versiones del entorno se tratará su implementación en sistemas tipo GRID u otros que no impliquen la limitación anteriormente mencionada.

5.4 Conclusiones sobre el entorno de diseño.

Como síntesis al capítulo, el ingeniero, que tradicionalmente se involucraba en cada una de las etapas del proceso de diseño, participa ahora exclusivamente en la especificación del problema, dejando al entorno la implementación automática del proceso y al usuario la evaluación subjetiva de las soluciones parciales obtenidas por él.

De este modo, el humano representa dos aspectos distintos dentro del proceso de diseño. Por un lado, define el “QUÉ” cuando se encarga de la formulación y análisis del problema. Y, por otro, el “CUAL” cuando interactúa con el sistema a través de una Interfaz Hombre-Máquina que le permite evaluar subjetivamente las soluciones parciales obtenidas. Del “CÓMO”, papel que anteriormente desempeñaba el humano, se encarga el entorno de diseño que sólo necesita de una mínima reconfiguración (operadores genéticos, decodificación, y encapsulamiento) cuando se implementa en él un problema específico.

5.5 Experimentos y Aplicaciones

La necesidad de liberar al humano de la fase de toma de decisiones no subjetivas que forma parte de todo proceso de diseño es acuciante cuando se trata con sistemas de gran complejidad. En este tipo de problemas los parámetros que definen la calidad de sus posibles soluciones en el dominio del problema están acoplados entre sí de forma no lineal, apenas están relacionados con los correspondientes a sus subsistemas en el dominio de aplicación o solución tecnológica y requieren para su análisis de más de un ámbito de conocimiento. Es en este tipo de sistemas donde el entorno de diseño automático desarrollado puede resultar más ventajoso asimilando la función de búsqueda creativa de posibles soluciones y liberando al humano de sus funciones de diseñador mediante su asignación exclusiva para la labor de especificación del problema.

Con este propósito, en los próximos dos capítulos se demostrará la versatilidad del entorno modular expuesto implementando el proceso de diseño de sistemas pertenecientes a ámbitos completamente distintos y que encajan perfectamente en el perfil mencionado.

En el capítulo “Robots submarinos” se diseñarán controladores de estabilización y maniobra para vehículos sumergibles basados en redes neuronales. La dinámica de este tipo de vehículos y las dificultades del modelado de las interacciones existentes con el entorno marino da lugar a la necesidad de sintetizar controladores no lineales de gran complejidad. Este problema se abordará para dos tipos de sistemas submarinos. Por un lado se tratará el estudio preliminar del diseño de la estructura de control de un vehículo submarino autónomo destinado a tareas y operaciones a gran profundidad y en un entorno limitado. Estos diseños preliminares buscan la controlabilidad del AUV gobernado por impulsores (thrusters) en el marco de un proyecto de desarrollo de un vehículo de estas características que se está llevando a cabo por nuestro grupo de investigación en colaboración con la empresa DINAIN. Por otro lado se considera el control de trayectoria ante consignas de un catamarán submarino de pasajeros gobernado por medio de superficies de control (timones de dirección y profundidad) y una planta impulsora. El problema en este caso busca un conjunto de controladores que, además de cumplir con las consignas tenga en cuenta la comodidad de los pasajeros del vehículo. Este desarrollo se ha realizado en el marco del proyecto PROTECAS en colaboración con la empresa SUBIBOR.

El primer problema abordado consistió en la estabilización en posición (hovering) de un vehículo sumergible autónomo que lleva instalado un brazo manipulador que sea capaz de compensar cualquier fuerza ejercida sobre el sistema incluyendo las perturbaciones externas y las reacciones provocadas por el movimiento del brazo. Los controladores no lineales generados por el entorno actuarán sobre los propulsores del vehículo en función de las señales que reciben desde un sistema de detección de tal modo que la posición del submarino permanezca inalterable.

El segundo problema para el cual se ha aplicado el entorno se basa en el diseño de controladores neuronales para la maniobra (seguimiento de consignas de cabeceo, guiñada, velocidad y profundidad) de un catamarán sumergible. En este caso los controladores actuarán sobre el sistema de propulsión y los timones de profundidad y

dirección que incorpora el vehículo en función de los datos proporcionados por sus sensores y compensando la acción de fuerzas perturbadoras.

Estos dos primeros problemas abordan el diseño de controladores no lineales acoplados para la realización de maniobras complejas. El problema fundamental aquí, además de la dificultad de la realización de cada controlador individual para tener en cuenta todas las posibles perturbaciones y problemas que pueden surgir en el medio submarino y para tener en cuenta la dinámica de éstos vehículos, cuya respuesta ante actuadores cambia de manera muy extrema dependiendo de la velocidad, maniobra que se esté realizando, orientación del vehículo, etc., surge del hecho de que el funcionamiento global de los vehículos depende de la acción coordinada de múltiples controladores con respuestas acopladas debido a la dinámica no lineal del vehículo en el medio.

En el siguiente capítulo, "Plantas industriales", se pretende aplicar la versatilidad del entorno para el diseño de la disposición y control de maquinaria en plantas industriales. En ambos casos se optimizarán parámetros de carácter general que intervienen en la calidad de las soluciones adoptadas tales como la productividad que presentan una relación extremadamente indirecta con los correspondientes a los subsistemas que se manejan. Resaltar que esta es la premisa fundamental de este ejemplo, tratar con parámetros de calidad que están muy cerca del universo del usuario, pero que resultan muy indirectas para la realización de cada controlador individual sin tener que especificar la transformación. En este caso, buscando que los ejemplos fuesen ejemplos de aplicación a problemas reales en entornos industriales reales, el desarrollo se llevó a cabo en el marco de un problema de diseño de línea de barnizado de la empresa SANTOS Cocinas y Baños.

El primer problema abordado en este capítulo consiste en la optimización de la disposición de maquinaria de una línea de ensamblaje de dos subproductos complementarios que presentan cadencias distintas. El segundo problema, y ya en la implementación real, se basa en la aplicación de técnicas de hardware evolutivo para la síntesis de un controlador secuencial que gestione los transportes de entrada y salida de una línea de barnizado real y que permita optimizar la productividad minimizando al mismo tiempo el gasto energético del proceso. Los circuitos lógicos secuenciales que constituyen cada controlador de la planta están basados en puertas lógicas nand y circuitos flip-flop que se codifican en forma de árbol binario. El controlador solución actuará sobre la velocidad de los transportes de la línea en función de las señales que aporta el sistema de detección de entidades de la planta de forma que minimice la existencia de esperas o atascos en la misma.

Hasta ahora las implementaciones comentadas corresponden a sistemas de gran complejidad en donde el entorno de diseño automático se hace cargo del proceso de diseño liberando al humano de la búsqueda y evaluación de soluciones a partir de las especificaciones y los recursos disponibles. Aunque la ventaja de desligar al humano del proceso de diseño para dejar a cargo del entorno la búsqueda de soluciones es obvia, no ocurre lo mismo con la evaluación en determinados casos. Por un lado y tomando como punto de partida el concepto de ingeniería concurrente el diseñador puede participar, de manera no exclusiva, en la evaluación de determinadas soluciones mediante la aportación de sus conocimientos técnicos. Además existen otros casos en donde el humano deja de representar su rol de diseñador para hacerse con un nuevo perfil, el de usuario. En este sentido la presencia del humano en el entorno de diseño posibilita el estudio de los aspectos sociológicos y subjetivos que intervienen en la aceptación de un producto.

Siguiendo esta misma línea se ha planteado el diseño automático e interactivo de morfologías tridimensionales mediante la aportación al entorno de diseño de un nuevo módulo interactivo que permita la evaluación de formas 3D modeladas en un sistema CAD clásico. Este nuevo concepto posibilita que la aptitud de las soluciones parciales obtenidas sea función de la combinación de criterios técnicos y subjetivos haciendo que el entorno pase a ser definido como un sistema de “ayuda al diseñador” en contraposición con los sistemas de “ayuda al diseño” que caracterizan al software de CAD clásico. De este modo con este nuevo sistema de diseño sencillo y económico es posible diseñar productos que se adapten al máximo a los gustos y necesidades de un perfil determinado de usuarios cumpliendo al mismo tiempo las especificaciones funcionales y las normativas técnicas.

Robots Submarinos

6 Robots submarinos

Como primer ámbito de aplicación se diseñarán controladores de estabilización y maniobra para vehículos sumergibles basados en redes neuronales. La dinámica no lineal y multidimensional de este tipo de estructuras complejas y las dificultades del modelado de las interacciones existentes con el entorno marino da lugar a la necesidad de sintetizar controladores no lineales de gran complejidad. En este capítulo se describirá cómo aplicando la metodología desarrollada en esta tesis se posibilita la automatización del diseño extrayendo al humano del proceso debido a sus limitaciones para tratar con problemas de este tipo.

6.1 Vehículos submarinos

Los robots submarinos pueden ayudarnos a entender mejor el entorno marino, proteger los recursos del océano contra la contaminación y utilizarlos eficientemente para el bienestar humano. Sin embargo, aunque la tecnología actual ha permitido que los seres humanos realicen grandes logros, existe un gran número de problemas complejos debido a las características del entorno submarino que hacen del océano tremendamente difícil de explorar.

De estos vehículos submarinos disponibles actualmente los más comerciales son los Sumergibles Operados Remotamente (ROVs Remotely Operated Vehicles) pero su uso es limitado al igual que ocurre con los sumergibles tripulados debido a costes operacionales y problemas de seguridad. Aunque los AUVs tienen más usos potenciales y grandes ventajas en relación a los ROVs en la actualidad sólo se encuentran en activo unos pocos modelos. La mayoría de ellos son vehículos de investigación sin manipuladores y solamente algunos desarrollan su actividad en aguas profundas. Su capacidad funcional es, por tanto, básica. Puede verse una reseña de algunos sumergibles desarrollados desde 1951 en [Nash, 95].

En los últimos años han aumentado los esfuerzos en investigación para desarrollar vehículos sumergibles que superen los problemas causados por el entorno no estructurado y peligroso del océano y al mismo tiempo, dotarlos de cierta autonomía reduciendo al mínimo la necesidad de la presencia de operadores humanos. El aumento de la actividad en I+D en este campo relacionada con el desarrollo de nuevos materiales y los avances en computación avanzada y tecnología sensorial nos conducirá en el futuro a vehículos submarinos completamente autónomos, especializados y fiables.

Además de la diferenciación fundamentada en la autonomía de los sumergibles, existen otros criterios como el sistema de propulsión y la misión a realizar que caracterizan cada uno de los vehículos sumergibles existentes. A continuación se expone una clasificación de estos artefactos y sus posibles aplicaciones.

6.1.1 Clasificación de los sumergibles

Los sumergibles se pueden clasificar en función de criterios técnicos y operativos.

- Clasificación en función de la existencia o no de tripulación

En el caso de los sumergibles tripulados el personal técnico que se aloja en su interior se encarga del control manual de la maniobra de la nave o de cualquier otra actividad relacionada con las operaciones a realizar tales como la investigación científica, instalación, inspección, mantenimiento y reparación de estructuras a poca profundidad. Además el cometido de la tripulación puede consistir en la supervisión de operaciones o incluso en la mera observación del entorno.

En cambio, los Vehículos Sumergibles No Tripulados o UUV (Unmanned Underwater Vehicles) son, literalmente, robots submarinos y están desplazando completamente a los vehículos tripulados. Las ventajas y desventajas operativas de un vehículo no tripulado frente a un vehículo tripulado son evidentes: por un lado se elimina el riesgo que supone hacer descender a una persona a gran profundidad (altas presiones, tiempos de permanencia limitados, dificultades de orientación, fauna peligrosa y fuertes corrientes entre otros), y por otro, se disminuye la habilidad y conocimientos intrínsecos que aporta el trabajar directamente sobre el terreno.

Sin embargo, a diferencia de cualquier otro tipo de artefacto no tripulado, un UUV se enfrenta con un problema particular especialmente exigente: las ondas electromagnéticas no penetran en el agua, razón por la cual los sistemas de control remoto convencionales son inútiles en el medio subacuático. Dada esta imposibilidad física, sólo quedan dos alternativas para maniobrar el aparato: los vehículos submarinos controlados remotamente y los autónomos.

Los Sumergibles Controlados Remotamente o ROVs (Remotely Operated Vehicles) se mantienen ligados a su embarcación nodriza mediante un cable o cordón umbilical de fibra óptica que permite enviar órdenes y recibir señales. En este caso, las decisiones son tomadas por el operador en superficie y todas ellas dependen de su destreza. Las aplicaciones de este tipo de artefactos se centran en la industria petrolera y de gas y en tareas de recuperación y/o rescate. En el campo de la ciencia su uso es similar al de los sumergibles tripulados.

Los AUVs a diferencia de los controlados remotamente, no dependen de ningún control en superficie, de forma que la única relación con ella se realiza por medios electroacústicos. La sensorización, control y actuación se realiza íntegramente en el robot que debe ser capaz de tomar decisiones en un ambiente desconocido. Hoy en día estos vehículos están cobrando gran importancia en la realización de distintas tareas (sobre todo en el campo de las aplicaciones offshore) a profundidades relativamente grandes.

- Clasificación en función de sus sistema de propulsión

La mayoría de los vehículos submarinos son Sumergibles Propulsados y se mueven horizontal y verticalmente auxiliados por thrusters. Sin embargo existen otros tipos de propulsión:

Los Sumergibles Autopropulsados son capaces de moverse siguiendo un cable guía situado en el fondo. Estos vehículos aplican un chorro de agua en el fondo marino para la excavación de la trinchera donde se instalarán cables submarinos y aprovechan la fuerza de reacción del chorro como propulsión. Este tipo de vehículo es poco común.

Los Sumergibles Remolcados son capaces de realizar pequeños movimientos para orientarse respecto al fondo, al buque de superficie u otro sumergible propulsado que lo remolca. Se dedican a tareas de inspección o para la excavación de zanjas y enterramiento de cables y tuberías de gran diámetro.

Los Sumergibles Autoremolcados se mueven de dos formas. Siguiendo un cable preinstalado en el fondo mediante un cabestrante, o la propia tubería a enterrar. El método de sujeción a la tubería se realiza mediante el uso de unas ruedas especiales.

Los Sumergibles tipo Oruga se desplazan reptando por el fondo marino. Son vehículos muy parecidos a los que operan en tierra pero preparados para operar bajo el agua. Pueden realizar múltiples tareas. Muchos de ellos llegan a pesar más de 30 toneladas. Entre sus actividades destacan realizar zanjas en fondos duros o rocosos para la instalación de tuberías.

- Clasificación en función de su sistema de maniobra

Los vehículos submarinos se clasifican también en función del sistema que utilizan para la realización de sus maniobras. Así, existen sumergibles que hacen uso de thrusters para su movimiento, los hay que utilizan planos o superficies de control a modo de timón y los hay que combinan ambas técnicas. Para la realización de los experimentos que veremos en este capítulo se modelaron vehículos de ambos tipos, un AUV y un catamarán.

- Clasificación en función de su tarea

Los sumergibles pueden clasificarse en seis clases diferentes en función de la misión para las que están encomendados.

Los Vehículos de Inspección y Observación o de Clase I son de pequeño tamaño y están equipados con videocámaras, focos y thrusters. Estos vehículos no suelen estar preparados para la instalación de aparatos auxiliares que les permitan realizar otro tipo de tareas.

Los de Clase II o Sumergibles de Trabajo se dividen en tres grupos Heavy, Medium y Light work. Estos vehículos son de mayor tamaño, peso y potencia que los anteriores. Normalmente están equipados con herramientas y manipuladores submarinos (brazos articulados con o sin pinzas). Además están preparados para que se les instale equipos

adicionales que les permiten realizar una gran variedad de trabajos. Obviamente también van equipados con cámaras y todo tipo de sensores que permitan el control máximo de la operación por parte del operador.

Los sumergibles de Clase III son Sumergibles de Bajo Coste, los de Clase IV se denominan Sumergibles de Propósito Especial, los Sumergibles Militares definen los de Clase V y por último los de Clase VI son Vehículos Sumergibles Autónomos (AUVs).

Esta variedad existente da idea del interés que suscita la explotación y exploración del mundo submarino y de la dificultad que supone el diseño y control de vehículos sumergibles. Que el entorno en donde se desenvuelve cada tipo de artefacto sea tan complejo y peligroso deriva en la especialización de su diseño en función de la misión encomendada.

6.1.2 Aplicaciones de los vehículos sumergibles

De todos los vehículos sumergibles las aplicaciones de los vehículos no tripulados o UUV (Unmanned Underwater Vehicles) cobran especial importancia en la actualidad por el extenso campo de aplicación al que pueden hacer frente. La mayoría de las operaciones de este tipo de vehículos son de origen científico y están orientadas principalmente al campo de la geología (estudio de la actividad sísmica, características de los fondos, recolección de muestras,...) y de la biología (estudio de fuentes hidrotermales, análisis de aguas, limpieza, fauna marina,...). Además existen otras actividades en las que toman parte como son las investigaciones pesqueras, hidrográficas, oceanográficas e incluso arqueológicas.

Los sumergibles no tripulados, además, han sustituido a los tradicionales buques oceanográficos como plataformas de observación debido al bajo coste de utilización y a que son capaces de medir autónoma y simultáneamente distintas variables físicas, químicas y biológicas con una gran resolución espacio-temporal. Además de este tipo de actividades la aplicación de estos vehículos es crucial en la inspección de estructuras sumergidas (presas, muelles, túneles de conducción, cables submarinos, tramos sumergidos de oleoductos y cascos de buques,...), búsqueda y localización de naufragios, ayuda en profundidad de embarcaciones submarinas, operaciones de inspecciones offshore (apoyo a la perforación y a la limpieza estructural), cableado energético y de telecomunicaciones, estructuras submarinas, actividades de uso militar, etc.

Poco a poco los submarinos no tripulados van dejando atrás a los tripulados en casi todo tipo de misión. Sin embargo el magnetismo y la belleza del océano ha dejado paso a nuevas actividades para vehículos sumergibles tripulados que aunque menos frecuentes se merecen un hueco en este apartado no sólo por su importancia creciente sino porque varios de los controladores sintetizados por el entorno de diseño desarrollado en esta tesis fueron implementados en un vehículo de este tipo. Me refiero a sumergibles de carácter turístico que además de abrir un nuevo campo de negocio en un entorno lleno de recursos como es el mar, incorpora nuevas restricciones debidos a los requerimientos de seguridad y comodidad de los pasajeros.

En cualquier caso, la consecución de cualquiera de las misiones enumeradas anteriormente supone un gran reto tecnológico debido a que el mar es un entorno hostil y variable que interacciona con estos vehículos perturbando seriamente el desarrollo de su misión. Como consecuencia de este comportamiento dinámico imprevisible

altamente no lineal y de gran incertidumbre surgió la necesidad de apoyarse en resultados obtenidos de la experimentación. La inviabilidad de la práctica real debido a su dificultad, peligrosidad y coste económico ha provocado el empleo de modelos hidrodinámicos menos complejos pero indudablemente más seguros. En el capítulo siguiente se abundará en el diseño y modelado de este tipo de vehículos.

6.2 Diseño y modelado de vehículos submarinos.

Existe un cuello de botella crítico en el diseño y desarrollo en robótica submarina y en especial en los Vehículos Submarinos Autónomos. A pesar de una gran cantidad de Sumergibles Operados Remotamente y una investigación rica en robots autónomos, existen pocos AUVs y sus capacidades son limitadas. El diseño morfológico de robots submarinos autónomos y el de sus controladores es particularmente difícil debido a los desafíos que supone la dinámica y la detección en un entorno submarino. Tanto es así que el comportamiento inesperado de la interacción existente entre los procesos del robot y el entorno actualmente se determina con la experimentación.

Además de esto, existe un gran hueco entre los proyectos teóricos y la práctica real en el diseño de robots submarinos debido a que la experimentación práctica es difícil, tediosa, infrecuente y potencialmente peligrosa. La evaluación significativa de resultados es obstaculizada por la complejidad del problema, insuficiencias en los sensores y la necesidad de observación remota. A pesar de estos impedimentos, hoy en día, está aumentando la necesidad de las operaciones submarinas debido a los recursos existentes en el suelo marino y a propósitos militares, de investigación y de rescate. Aunque la investigación realizada sobre los robots submarinos ha sido lenta y gradual, la existencia de estos desafíos fuerza a los diseñadores a proyectar robots robustos, autónomos, móviles y estables.

En comparación con otros entornos, el dominio submarino impone ciertas limitaciones para el estudio de la dinámica, control, comunicación y sensorización de todo vehículo submarino. Tanto es así que el diseño y la construcción de vehículos submarinos se ocupa casi totalmente de los factores del entorno que interactúan entre sí y que afectan al comportamiento dinámico de un vehículo sumergible.

El océano rodea totalmente el vehículo, afecta en gran medida a su movimiento y plantea un peligro constante para la supervivencia del mismo. Muchos de los efectos del entorno circundante en un robot son únicos al dominio submarino y los vehículos que se mueven en él intentan controlar fuerzas y reacciones complejas de una manera fiable y predecible. Entender estas fuerzas es un requisito importante en el diseño y control de los comportamientos de este tipo de vehículos. Estos factores pueden dar lugar a un comportamiento oscilatorio o inestable si los algoritmos de control de dirección, profundidad y velocidad no tienen en cuenta las muchas respuestas complejas posibles del vehículo.

Un AUV en funcionamiento es inaccesible, remoto y desatendido, está sujeto a valores extremos de presión, temperatura y corrosión y las comunicaciones con el exterior son intermitentes o inexistentes. El despliegue, la operación y la recuperación del vehículo de sólo algunas horas de autonomía requieren mucho tiempo y son costosos y el sistema de propulsión es caro, lento y limitado. Debido a este entorno poco amistoso el control dinámico de un vehículo de este tipo supone muchas veces un desafío.

Un entorno de trabajo duro y una susceptibilidad al fallo físico están entre las razones principales del bajo número de proyectos y diseños de este tipo. El fallo de un AUV en el océano es inaceptable por varias razones: cualquier fallo puede llegar a ser catastrófico, la recuperación es difícil y a veces imposible y los costes de reemplazo en tiempo y dinero son prohibitivos.

Además, el movimiento del vehículo sumergible no se restringe a la dirección vertical y las restricciones del mismo sufren grandes variaciones durante la interacción con diversos entornos. Para algunos vehículos no tripulados, el movimiento debe ser restringido para alcanzar ángulos suaves de cabeceo (inclinación respecto a su eje estribor-babor), balance (rotación respecto al eje longitudinal popa-proa) y guiñada (giro en torno al eje cubierta inferior-superior). Esta restricción se impone debido a que un movimiento brusco puede causar daños en los equipos y en la tripulación debido a la posibilidad de ángulos de ataque muy grandes entre la orientación del vehículo y la dirección del movimiento del mismo.

Podemos concluir que en el diseño de un AUV la fiabilidad, la estabilidad y la autonomía son cruciales puesto que las restricciones de un AUV son a menudo las peores para cualquier tipo de robot debido a los desafíos inherentes del entorno submarino. Esta vulnerabilidad crítica en el diseño de un vehículo submarino es una causa que contribuye a explicar la razón del escaso trabajo realizado sobre ellos.

La dificultad que entraña el diseño de vehículos submarinos se concreta a la hora de detallar su estructura, su sistema de potencia, comunicaciones y sensorización y los sistemas que a continuación se indican.

La respuesta dinámica de un vehículo sumergible depende al mismo tiempo del diseño de la morfología exterior y del tipo, disposición y control de su sistema de propulsión. Es por eso por lo que el diseño de ambos sistemas están intrínsecamente relacionados siendo deseable que su modelado y optimización sea conjunta para la síntesis de algoritmos de control de navegación automática.

6.3 Control de vehículos submarinos.

El desarrollo de los AUVs comenzó a principios de los años 60 como evolución natural de los ROVs (Remotely Operated Vehicles) y han progresado a un estado donde pueden ser alternativas comercialmente viables. A diferencia de los ROVs que realizan sus operaciones ayudándose de manipuladores en un estado inmóvil o a muy baja velocidad, los AUVs deben poder funcionar realizando tareas complejas y misiones diversas durante extensos períodos de tiempo y realizar sus misiones sin la intervención humana mientras mantienen un control estable de seis grados de libertad. Para ello necesitan de un controlador para su posicionado y movimiento dinámico y de muchas clases de sensores que les permitan explorar las profundidades del océano a alta velocidad de crucero.

Los controladores diseñados para un AUV deben posibilitar la realización de operaciones de "homing" y "docking" si e. que se necesita una autonomía total. Homing es una operación que hace retornar al AUV a las proximidades del barco nodriza una vez realizada la misión que se le ha encomendado y Docking es otra operación que usa el AUV para situarse en una misma posición cuando está cerca del buque. Ambas operaciones requieren de una planificación y control de la trayectoria prevista, pero la operación de Docking es la que presenta mayor dificultad debido a que

la posición del buque nodriza puede cambiar fácilmente debido a las corrientes y a las olas del océano. Por lo tanto, se necesita una trayectoria mucho más exacta que considera la posición futura del buque. Es precisamente esta última operación la que pretende realizar un AUV con manipulador con la ayuda de los controladores sintetizados con el entorno de diseño automático presentado en esta tesis.

Los aspectos más importantes que hacen difícil de controlar los robots submarinos incluyen: alta no linealidad, comportamiento dinámico variable con el tiempo, incertidumbre en los coeficientes hidrodinámicos, estructura redundante y de mayor orden cuando se une a un manipulador, existencia de perturbaciones debidas a las corrientes del océano y cambios en los centros de gravedad y flotabilidad debido al movimiento del manipulador que también perturba el cuerpo principal del robot. Por consiguiente, el sistema de control para AUVs se convierte en un conjunto complejo que debe poder adaptarse a entornos cambiantes e imprevisibles y al mismo tiempo realizan su tarea asignada. Ante tal panorama sería deseable obtener un sistema de control con capacidad de auto ajuste debido a los cambios en la dinámica del robot y de su entorno. Este análisis y diseño de controladores robustos se centra en producir un funcionamiento estable al controlar sistemas multivariable con incertidumbre significativa [Dorato, 87].

Se han propuesto muchas estructuras de control para los vehículos submarinos. En [Goheen, 90] los autores desarrollaron un controlador cuyo objetivo era realizar el auto posicionamiento y el mantenimiento en posición "autoposition and station holding". Este tipo de problemas son muy difíciles de abordar si se usan metodologías convencionales de sistemas y esquemas de control debido a la carencia de modelos precisos y de la incertidumbre del entorno hidrodinámico no estructurado en el cual funcionan los AUVs. Algunos autores han recurrido al desarrollo de prototipos de AUVs como entornos de software de control (ejemplos de esto son ODIN en la Universidad de Hawaii [Choi, 95], Twin-Burger en RIKEN [Fujii, 96] o SAUB en el Autonomous Undersea Systems Institute (AUSI) [Patch, 00]), pero éste es un proceso muy largo y costoso y los sistemas obtenidos de esta manera no se pueden generalizar fácilmente a otras plataformas con excepción de en las que fueron diseñados.

También se han realizado varios métodos para el control de tolerancia a fallo "fault-tolerant" de robots submarinos autónomos [Babcock, 90], [Healey, 92], [Dunn, 92], [Orrick, 94], [Takai, 94], [Yang, 99]. Para cualquier fallo importante de los subsistemas del robot, el robot debe emerger a la superficie y señalar su situación para su recuperación, sin embargo, para cualquier fallo tolerable, el robot debe ajustarse al fallo y terminar la tarea asignada. Un sistema tolerante a fallo eficiente y eficaz llega a ser imprescindible para un AUV y consiste en tres áreas: detección de avería, aislamiento de fallos y ajuste a la avería.

En la bibliografía se pueden encontrar diversas aproximaciones al control de maniobra de vehículos submarinos, aunque es difícil encontrar una referencia en la que se efectúe el control en los 6 grados de libertad debido a la dificultad que conlleva el acoplamiento de controladores. Resulta bastante común diseñar un controlador lineal para el vehículo como un autopiloto partiendo de un modelo lineal del mismo. Este modelo se desarrolla a partir de un punto estacionario de operación obteniendo así la función de transferencia como han hecho Zhengping Feng y Robert Allen [Feng, 01]. Este modelo habría que verificarlo comparando las respuestas del modelo linealizado, obtenido a partir del no lineal, con el real no lineal bajo las mismas perturbaciones de las entradas de control. Obtener el modelo linealizado es un trabajo arduo cuya validez es, cuando menos,

cuestionable en casos como el que nos ocupa, donde las respuestas son altamente no lineales y donde existen diversos procesos acoplados. Algunos autores han recurrido por ello al desarrollo de prototipos como elementos de prueba del software de control. Un ejemplo de esto es ODIN en la Universidad de Hawaii [Choi, 95].

Se han diseñado sistemas de control submarinos avanzados para robots, tales como:

- Control en modo “sliding”

Yoerger y Slotine [Yoerger, 85] han propuesto un control de trayectoria investigando los efectos de la incertidumbre de los coeficientes hidrodinámicos y de la eliminación de los términos de acoplamiento.

Healey y Lienard [Healey, 93] han dividido al sistema en subsistemas independientes de movimiento, inmersión y velocidad.

- Control no lineal

Otro tipo de sistemas de control para vehículos submarinos, con mejores resultados que los controles PD y PID, son los basados en modelos no lineales, consistentes en un modelo de planta dinámico y una trayectoria de referencia. Las aproximaciones dimensionales finitas para la dinámica del vehículo submarino son estructuralmente similares a las ecuaciones de un sólido rígido. A este modelo se aplican técnicas de control adecuadas al modelo, evaluando la estabilidad analíticamente. Un ejemplo de métodos de control para este tipo de plantas dinámicas es el aplicado David A. Smallwood y Louis L. Whitcomb [Smallwood, 01].

Nakamura y Savant [Nakamura, 92] han propuesto un control de seguimiento de trayectorias no lineal de un AUV de 4 grados de libertad considerando el movimiento cinemático haciendo uso de la naturaleza no holonómica del sistema.

- Control adaptativo

Goheen [Goheen, 90] propuso controladores autoajustables multivariables tipo piloto automático para que los vehículos submarinos superasen las incertidumbres del modelo en el autoposicionamiento y el mantenimiento en posición.

Choi y Yuh [Choi, 96] han desarrollado un controlador adaptativo que ha sido implementado para la navegación de un AUV Omni Directional Intelligent Navigator (ODIN) en. Un control adaptativo híbrido (que sugiere que el procedimiento sea una mezcla de operaciones continuas y discretas) de un AUV fue desarrollado por Tabaii en [Tabaii, 94]. El sistema fue simulado en el dominio continuo mientras que las secciones de control y de identificación eran discretas.

En numerosas referencias [Yoerger, 85, 90], [Papoulias, 89, 91], [Cristi, 91, 95], [Healey, 92, 93, 94], [Fossen, 94] se analizan controladores sofisticados entre los que destacan los basados en métodos de control adaptativos y aplicaciones de técnicas de aprendizaje [Goheen, 87].

Para abordar directamente el problema de la no linealidad utilizando modeladores/controladores intrínsecamente no lineales, algunos autores han recurrido a sistemas basados en redes neuronales artificiales o a sistemas neuronales-borrosos.

- Control con redes neuronales

En el trabajo de V.S. Kodogiannis, se describe la aplicación de las redes neuronales para el control de un vehículo submarino operado remotamente [Kodogiannis, 01]. En dicho trabajo Kodogiannis combina el control adaptativo con las arquitecturas de redes neuronales para aproximar la no linealidad del sistema y su dinámica a la simulación del vehículo.

Yuh [Yuh, 90, 94] ha propuesto un sistema de control neuronal usando un algoritmo de aprendizaje reforzado para que el sistema se ajuste directamente y en línea sin un modelo explícito de la dinámica del vehículo.

Lorentz y Yuh en [Lorentz, 96] presentan resultados experimentales en ODIN usando el método propuesto por Yuh.

Ishii propuso en [Ishii, 98] un sistema de control basado en una red neuronal llamado "Self-Organizing Neural-Net-Controller System" (SONCS) para AUVs y examinó su eficacia con el uso para el control de mantenimiento de cabeceo de un AUV llamado "Twin-Burger". En su estudio, un método rápido de adaptación del controlador llamado "Imaginary Training" se utiliza para mejorar el proceso de consumo del tiempo de adaptación de SONCS.

- Control borroso

Otros autores, tales como Mills y Harris [Mills, 94] o como Suto y Ura [Suto, 93] usan sistemas neuro-borrosos o simplemente conjuntos de redes neuronales diseñadas ad hoc para sus características particulares utilizando sistemas basados en aprendizaje. El problema con estas aproximaciones es la necesidad de disponer de un conjunto de entrenamiento y éste es muy difícil de obtener individualmente para cada controlador cuando los diversos sistemas de control están acoplados. El diseño manual de dichos conjuntos de controladores se vuelve, en muchos casos inabordable.

DeBitetto [DeBitetto, 94] desarrolló un controlador de profundidad borroso de 14 reglas para un UUV.

Tsukamoto [Tsukamoto, 99] implementó experimentalmente cuatro controladores de posición y velocidad para un sistema de thrusters: una red neuronal on-line, un controlador neuronal off-line, un controlador borroso y otro adaptativo.

También se han definido arquitecturas de control para vehículos submarinos autónomos [Valavanis, 97], [Girard, 98].

- Control evolutivo

La evolución de controladores de guiñada, profundidad, cabeceo, tracking y hovering, etc. basados en la implementación de algoritmos genéticos se hace complicada pero es factible dada la complejidad del diseño de sistemas de control en este tipo de vehículos. La posibilidad de probar repetidamente en un simulador los controladores sintetizados por el algoritmo haciendo uso del modelo hidrodinámico anteriormente definido hacen de este último una herramienta imprescindible para la investigación en este terreno. Este es un proceso, en todo caso, carente de riesgo para el robot, el humano y el entorno pero

que necesita de un elevado tiempo de procesamiento computacional y del auxilio, como consecuencia, de la programación distribuida.

En esta memoria se propondrá una aproximación de este tipo para la obtención automática de controladores de navegación y estabilización de AUVs en entornos cambiantes pero sin considerar modelos extremadamente precisos de perturbaciones hidrodinámicas y sin recurrir a prototipos para el ajuste fino de los controladores. La generalidad del proceso de síntesis de los controladores y la robustez de los mismos serán analizados más adelante.

Además, el entorno de diseño desarrollado posibilita la evolución conjunta del diseño de la morfología y control de este tipo de vehículos. Esto supone una auténtica innovación en este campo debido a la dificultad que entraña su implantación.

A continuación se describirá la aplicabilidad de la simulación para el diseño de sumergibles y su modelado hidrodinámico y sensorial.

6.3.1 Simulación

La dificultad del mundo submarino plantea la necesidad de recurrir a un entorno práctico, seguro y completo para la experimentación hidrodinámica de AUVs. Debido a que los canales de ensayo hidrodinámicos existentes no pueden reproducir la variabilidad de los parámetros críticos encontrados en el océano, el modelado tridimensional por ordenador se hace un hueco para intentar describir las características funcionales principales del entorno real, permitiendo la evaluación más o menos realista del AUV en el laboratorio. El comportamiento físico de la dinámica del robot debe ser reproducido correctamente, puesto que los vehículos submarinos son propensos a las inestabilidades dinámicas no lineales y a respuestas físicas impredecibles que pueden dar lugar a la pérdida del vehículo.

Se ha producido gran cantidad de simulaciones relacionadas con la robótica, pero pocas relacionadas con la robótica submarina. Esas simulaciones que están disponibles son restringidas normalmente por limitaciones comunes de la simulación. Así, los resultados de la simulación siguen siendo susceptibles a fallos cuando son evaluados en el mundo real debido a la complejidad del comportamiento dinámico del sistema.

La construcción de un entorno de simulación para el desarrollo del robot y la evaluación del comportamiento del mismo es un requisito previo necesario para el diseño de un robot complejo que opera en un entorno arriesgado, como el de un AUV. En él se recrea cada aspecto del entorno externo del robot así como los sensores del robot y los dispositivos analógicos de maniobra (thrusters y timones). Estos entornos de simulación dan la capacidad a los robots de interactuar con los entornos virtuales y superar los handicaps del diseño inherentes en la construcción de un robot submarino que opera en un entorno peligroso. Los entornos virtuales submarinos pueden romper el cuello de botella presente en el desarrollo de los AUVs.

6.3.2 Modelos hidrodinámicos

El modelado de la respuesta hidrodinámica del movimiento del vehículo sumergible es esencial debido a la necesidad de evitar posibles errores en el sistema de control, sensorización, navegación y propulsión. Cualquiera de estos errores puede ocasionar la

pérdida del control del vehículo e incluso el fallo catastrófico debido a la colisión con el fondo marino.

El modelo hidrodinámico de un vehículo submarino debe reproducir exactamente su respuesta a las fuerzas y órdenes de control que se apliquen sobre él y ser, al mismo tiempo, general, verificable, parametrizable y conveniente para la simulación distribuida. El modelo debe ser lo suficientemente exacto respecto al comportamiento equivalente real bajo el agua para permitir el diseño y la evaluación realistas y repetibles de los sistemas de control diseñados para el vehículo. La inclusión de un modelo analíticamente correcto y verificable de la hidrodinámica en un entorno de simulación submarino permitirá, además, el análisis significativo y oportuno de las interacciones realistas entre el robot y el entorno.

El objetivo de un modelo hidrodinámico es determinar el estado del sistema, es decir, el mínimo conjunto de variables a partir de las cuales se puede determinar de forma única la posición, la orientación y la energía cinética y potencial del vehículo.

El examen de las características generales de los modelos dinámicos de un robot en diferentes entornos revela que el caso submarino es muy difícil de especificar con exactitud, el más difícil de verificar y el más crítico para prevenir la pérdida catastrófica del vehículo. Tan alta complejidad es debida a los múltiples efectos del entorno circundante en forma de fuerzas y momentos acoplados en todas direcciones que son tan significativos que el agua adyacente tiende a ser acelerada junto con el vehículo. Este concepto se conoce como "masa añadida".

Hay cientos de factores correlacionados relativos a los efectos lineales y no lineales de la elevación, fricción, masa añadida y propulsión en este tipo de modelos. Aunque alguno de estos coeficientes son efectos de segundo orden o de importancia insignificante, la determinación de los coeficientes primarios es muy difícil y costosa. En general, los parámetros hidrodinámicos que modelan la dinámica de un robot submarino son altamente no lineales, acoplados y dependientes del tiempo y su dificultad de manejo se incrementa cuando el vehículo presenta una estructura abierta con superficies irregulares.

Todo esto hace que el modelado de la respuesta de un vehículo submarino se convierta en un problema extremadamente difícil de dinámica. Tanto es así que la carencia de modelos apropiados ha obstaculizado en gran medida el desarrollo de investigaciones en este campo. Actualmente no existe un modelo general y único para la hidrodinámica de movimiento (propulsores, planos) y de estabilización (thrusters) de un vehículo submarino. Aunque existen numerosos artículos sobre la hidrodinámica subacuática, casi sin excepción son modelos parciales de dinámicas subacuáticas que se centran en ciertos aspectos y sufren de incompletitud, nomenclatura confusa, demasiada simplificación, o de una formulación inadecuada.

Además, si al vehículo se le incorpora un cable umbilical se añaden al modelo fuerzas y momentos significativos que incrementa la complejidad de las ecuaciones hidrodinámicas del movimiento del mismo [Abel, 72], [Brancart, 94], [Hover, 94]. La complejidad del modelado de las olas y corrientes del océano es también alta y se extiende desde sinusoidales simples a modelos numéricos sofisticados ejecutados en superordenadores que analizan datos empíricos [Covington, 94], [Fossen, 94], [Musker, 88], [Blumberg, 94]. El principio de superposición permite que los efectos de las olas y las corrientes sean añadidos a la solución hidrodinámica al final del algoritmo,

implicando que los modelos altamente complejos de las olas y las corrientes del océano pueden ser solucionados off-line o en paralelo. La detección de colisiones es otro problema con un grado extremadamente alto de complejidad de cómputo debido que los cambios abruptos en la aceleración y la velocidad del cuerpo pueden requerir una revisión del algoritmo temporal de integración.

Existen numerosos textos sobre la dinámica de vehículos marinos, el más notable es el de [Lewis, 88], pero su estudio se concreta casi exclusivamente en las naves de superficie. [Jackson, 92] define también un buen compendio pero en este caso sobre de los fundamentos del diseño submarino naval. Se encuentran más detalles sobre modelos hidrodinámicos de submarinos en [Huang, 88].

Algunos investigadores que trabajan en el tema han propuesto varias técnicas de modelado e identificación de sistemas para los vehículos robóticos submarinos [Fossen, 95], [Goheen, 95]. Entre ellos destaca el modelo hidrodinámico más riguroso y casi completo de Healey en [Healey, 93], que se describe en [Fossen, 94] y que llena un hueco significativo en la literatura de la robótica y de la simulación.

Ejemplos de modelos hidrodinámicos de vehículos submarinos son:

- Healey presentó un modelo inicial de la hidrodinámica de un vehículo submarino [Healey, 92, 93] que definió la base fundamental para las investigaciones en este campo. El modelo incluyó además de los coeficientes hidrodinámicos obtenidos empíricamente, las fuerzas que intervenían en las operaciones de maniobra de un vehículo controlado por planos, timones y propulsores. En el modelo fueron pasados por alto los términos relativos a las fuerzas y los momentos de los thrusters, las fricciones a poca velocidad, las configuraciones inusuales del vehículo y las ecuaciones diferenciales no fácilmente adaptadas para la integración.
- Fossen en 1994 publicó un tratado sobre la dinámica y control de vehículos submarinos y de superficie donde se estudió la estabilidad, el modelado del océano, del viento y de las olas y técnicas avanzadas del control [Fossen, 94].
- El equipo de diseño y desarrollo de Navy/ARPA UUV divulgó un modelo completo de la hidrodinámica de seis grados de libertad para el desarrollo y test de controladores sofisticados para vehículos submarinos [Pappas, 91] [Brancart, 94].
- Yuh en [Yuh, 90] presentó un modelo de un ROV válido para cualquier tipo de vehículo submarino. Él describe la "masa añadida" y la mayoría de los otros términos relevantes.

Existe poca bibliografía no clasificada referente a los modelos hidrodinámicos de vehículos submarinos de la Marina de los Estados Unidos. [Jackson, 92] proporciona una descripción del proceso básico del diseño submarino, mientras que [Gertler, 67], [Feldman, 79] y [Lewis, 88] presentan de forma general las ecuaciones dinámicas y la nomenclatura de los coeficientes.

El resolver un problema de mecánica de fluidos mediante aproximaciones matemáticas con la ayuda del ordenador constituye la base de los denominados CFD, siglas de la expresión inglesa "Computational Fluid Dynamics", Dinámica de Fluidos Computarizada y que se encuentra en fase de desarrollo. No existen apenas soluciones de este tipo y los métodos numéricos que procuran solucionar las ecuaciones

diferenciales parciales de Navier-Stokes tienden a introducir parámetros muy complicados que luego se eliminan debido a su gran complejidad para el cómputo [Draper, 94]. Esto es un problema para el modelado de AUVs que tienen a menudo formas irregulares. Los métodos CFD, por tanto, no son actualmente convenientes para la simulación de la hidrodinámica subacuática del vehículo.

6.3.3 Simuladores

A continuación se describen los principales simuladores reales, puramente computacionales o híbridos que se han utilizado para investigación en este campo.

La simulación integrada desarrollada por Brutzman [Brutzman, 92] y Compton [Compton, 92] proporcionó un conjunto de herramientas de simulación que asisten en el diseño y simulación de vehículos submarinos y en el que es posible aplicar algoritmos relacionados con la Inteligencia Artificial. Como aplicación de la simulación integrada se desarrolló el simulador Multi-Vehículo (MVS) que se aplicó con éxito a un modelo del Twin-Burger AUV [Kuroda, 94] [Brutzman, 94].

El laboratorio de ARPA/Navy desarrolló un simulador híbrido consistente en un ordenador central, modelos de hidrodinámica y respuesta de sensores y modelos altamente detallados a nivel de componentes del equipo interno individual del UUV (tales como modelos electrodinámicos del motor) [Pappas, 91] [Brancart, 94] [Brutzman, 93, 94]. Una unidad de interfaz de simulación Simulation Interface Unit (SIU) proporciona una interfaz configurable entre el ordenador central y el vehículo. Los algoritmos y el código fuente para el simulador híbrido no están disponibles públicamente y muchos componentes del equipo son propietarios. Aunque la arquitectura del software no es escalable ni distribuida este simulador es uno de los mejores del tipo "hardware en el lazo".

El grupo IMG de la NASA ha trabajado en gran variedad de robots e interfaces hombre-máquina con el objetivo a largo plazo de proporcionar telepresencia eficaz para la exploración científica de otros entornos, como por ejemplo en Marte [Hine, 94].

El Omni-Directional Intelligent Navigator (ODIN) de la Universidad de Hawaii [Choi, 94] combina un AUV con una simulación gráfica integrada para el desarrollo de los algoritmos adaptativos de control [Choi, 94]. ODIN es un AUV esférico pequeño con un manipulador simple y cuatro propulsores verticales orientables, capaces de controlar la posición en seis grados de libertad. La investigación inicial con ODIN se usó para la determinación de los coeficientes hidrodinámicos, controladores lineales, no lineales y adaptativos usando detección de fallos y reconfiguración automática mediante redes neuronales.

Tuohy [Tuohy, 94] desarrolló un modelo de la simulación para probar aplicaciones de navegación de AUVs. El modelo total de simulación orientado a objetos y de propósito general permitía diseñar el vehículo y el entorno a partir de mapas tridimensionales.

Chen [Chen, 94] describe cómo la mayoría de las simulaciones de robótica incluyen el robot y el entorno mientras que excluyen los sensores y presenta un sistema para la simulación y la animación de manipuladores comandados por sensores y robots móviles de interior. Los gráficos interactivos tridimensionales se utilizan para la visualización del robot y de los sensores, aunque el funcionamiento en tiempo real no esté garantizado. Los robots se pueden integrar en el sistema de simulación para permitir el

funcionamiento en modo real o modo virtual. La conclusión primaria de este trabajo es que un simulador para un sistema robótico integrado comandado por sensores debe incorporar información sensorial.

El simulador de robots móviles Ars Magna de la Universidad de Yale, proporciona un mundo abstracto en el cual un planificador basado en Inteligencia Artificial y programado en LISP es capaz de controlar el movimiento cinemático de un robot móvil basado en un solo punto [Engelson, 92]. También se incluyen modelos de manipuladores y los valores de los sensores son proporcionados por los modelos geométricos con distribuciones ajustables de ruido. Las capacidades útiles pero limitadas del Ars Magna son representativas de la mayoría de los otros simuladores de robots actualmente en uso.

Por lo tanto, el modelado y la verificación exactos mediante simulación es un requisito esencial en el proceso del diseño [Lewis, 84], [Pappas, 91]. El uso de la simulación integrada con partes reales del vehículo y de su entorno (HILS: hardware in the loop simulation) sería preferible a la simulación independiente totalmente numérica. Sin embargo, los paquetes integrados de simulación con gráficos en 3D son más útiles para desarrollar vehículos submarinos avanzados puesto que la experimentación en campo es muy costosa [Choi, 1993], [Brutzman, 92], [Kuroda, 95].

6.3.4 Sensores

La información sensorial esencial para el éxito de una inmersión y la seguridad del vehículo es la profundidad e inclinación del mismo. Para medir estas magnitudes se hace uso normalmente de los sonars y los sensores electromecánicos e inerciales. Existen paquetes completos que usan sensores de estado sólido y de circuitos integrados a coste relativamente bajo que proporcionan valores angulares y de aceleración en los tres ejes cuerpo. Las velocidades se obtienen a partir de integración de las aceleraciones y los valores de la posición se calculan por la integración de la velocidad. Desafortunadamente la selección, compra, instalación, calibración, prueba e interpretación de sensores consume mucho tiempo y es caro.

La exactitud de dispositivos inerciales depende de la estimación del ángulo pitch y roll y la sensibilidad de la aceleración. Los acelerómetros inerciales son afectados por las aceleraciones en el vehículo y las aceleraciones debidas a la gravedad. Como la aceleración debida a la gravedad es aproximadamente diez veces mayor que la aceleración de los propulsores usados por vehículos de velocidad lenta, una estimación exacta de los ángulos del vehículo es esencial para aislar los componentes de aceleración únicos del vehículo. Debido a que las estimaciones de la posición y de la rotación son integraciones dobles de aceleraciones, cualquier ruido o error en la estimación de aceleración se amplifica grandemente en el transcurso de tiempo.

Un valioso beneficio de los modelos hidrodinámicos, como los aplicados en los casos prácticos que se expondrán más adelante, es que pueden proporcionar valores de "sensores virtuales" que pueden o no instalarse físicamente. Los "sensores virtuales" indican en cada instante los ángulos cabeceo, balance y guiñada del vehículo y son particularmente útiles al evaluar el control bajo condiciones de funcionamiento variables (especialmente simulando el fallo del sensor). La clave al éxito cuando se producen tales simulaciones estará incorporando modelos del error.

6.4 Diseño de Controladores para Vehículos Submarinos

La metodología evolutiva de diseño automático desarrollada en esta tesis combina la fidelidad de la simulación proporcionada por los simuladores avanzados mecánicos y de fluidos actuales con la flexibilidad y versatilidad de técnicas evolutivas y la potencia de la computación distribuida para producir una herramienta de diseño automática que pueda obtener diseños complejos que trabajan en el mundo real en comparativamente poco tiempo a costo muy bajo.

Obviamente, la utilidad de los resultados obtenidos será tanto mejor cuanto más precisos sean los simuladores utilizados para evaluar los fenotipos resultantes y, como es bien sabido, fidelidad en simulación normalmente resulta en alto coste de computación. Es por esta razón por la que éste es un entorno distribuido capaz de operar en clusters de ordenadores utilizando MPI.

Los dos pilares en que se sustenta la implementación práctica del entorno de diseño desarrollado para el desarrollo de controladores para vehículos submarinos son la capacidad de las redes neuronales artificiales para generalizar funciones no lineales complejas a partir de un conjunto discreto de ejemplos y la exploración eficiente del espacio de diseño que proporciona la utilización de Algoritmos Genéticos o Evolutivos. Estas dos metodologías se combinaron con un conjunto de simuladores hidrodinámicos y de estructuras de control para permitir la obtención de grupos de controladores apropiados.

La inclusión en el entorno de Redes Neuronales Artificiales (RNAs) se fundamenta en su capacidad de tolerancia a fallos y ruido (característica crucial debido a su presencia en entornos reales) y su capacidad de flexibilidad, versatilidad y generalización a partir de un sistema discreto de ejemplos (no en vano se le considera como un aproximante universal). Todas estas propiedades resultan especialmente atractivas para la implementación de las RNAs en robótica autónoma.

La razón de usar los algoritmos genéticos como método de búsqueda se debe a la dificultad que supone la determinación a priori de la mejor acción o secuencia de acciones que el robot debe realizar para la consecución de cualquiera de sus objetivos. Especialmente porque la acción puede implicar un compromiso entre varios casos perceptualmente idénticos para el robot que son en realidad diferentes. Además cuando la complejidad del comportamiento aumenta o cuando el nivel de ruido es tan alto que el diseñador no puede elegir la estrategia óptima, llega a ser muy difícil hacer uso del aprendizaje.

Así, los individuos de la población son los controladores candidatos para un comportamiento determinado del robot y los genes del cromosoma los parámetros que describen cada red neuronal. Estos parámetros se identifican generalmente con los pesos sinápticos aunque es posible que correspondan a parámetros adicionales como el número de capas, el número de neuronas por capa, el tipo de neurona, etc.

A continuación se presentarán dos aplicaciones prácticas del entorno de diseño para la síntesis de controladores de vehículos submarinos que consisten en el diseño de controladores de estabilización y maniobra para dos tipos de sumergibles completamente distintos: un AUV y un catamarán sumergible

6.4.1 Hovering

Este primer ejemplo trata de la realización de los diseños preliminares y el estudio de la problemática del control de un AUV que nuestro grupo está desarrollando en colaboración con la empresa DINAIN.

El objetivo de este ejemplo consiste en la síntesis de controladores para la estabilización en posición de un AUV al que es posible incorporar opcionalmente un brazo manipulador excavador.

Los controladores generados guiarán al sistema de propulsores en la maniobra de mantenimiento en posición independientemente de las fuerzas y las perturbaciones que se apliquen, incluyendo, en su caso, la perturbación generada por el movimiento del brazo y los cambios en su peso cuando excava. Es decir, el AUV deberá compensar cualquier fuerza ejercida sobre él (incluyendo la reacción del movimiento del brazo excavador si éste existe) de modo que su velocidad sea siempre cero en cualquier dirección y su posición permanezca inalterable.

El control diseñado actuará sobre la fuerza ejercida por los propulsores en función de las entradas de detección del AUV constituidas por los valores obtenidos mediante un sistema de acelerómetros que permite monitorizar las aceleraciones angulares y lineales del vehículo.

En la figura 6.1 se muestra una vista esquemática del vehículo en donde también se representan las fuerzas aplicadas que sobre él se aplican y la posición de los 6 propulsores que utiliza para su movimiento. El sistema presenta 6 grados de libertad con lo que puede girar alrededor de cualquier de sus ejes y desplazarse en cualquier dirección.

Para optimizar los controladores basados en RNA's se empleó un algoritmo genético cuya particularidad principal consiste en que la evaluación de la aptitud de los mismos se realiza por medio de un simulador dinámico comercial, DADS de CADSI. El proceso consistió en la exportación del modelo del AUV realizado en este software a un bloque en Simulink y la integración del mismo con los controladores diseñados. Simulink fue el encargado de enviar los comandos de control al simulador y recibir de éste los valores de los sensores del AUV de modo que el algoritmo genético pudiera calcular la aptitud de la solución según los criterios establecidos.

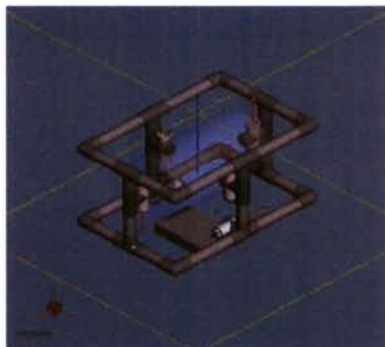


Figura 6.1: Modelo del AUV

En el siguiente gráfico se muestra el esquema funcional del Entorno Evolutivo de Diseño Automático desarrollado.

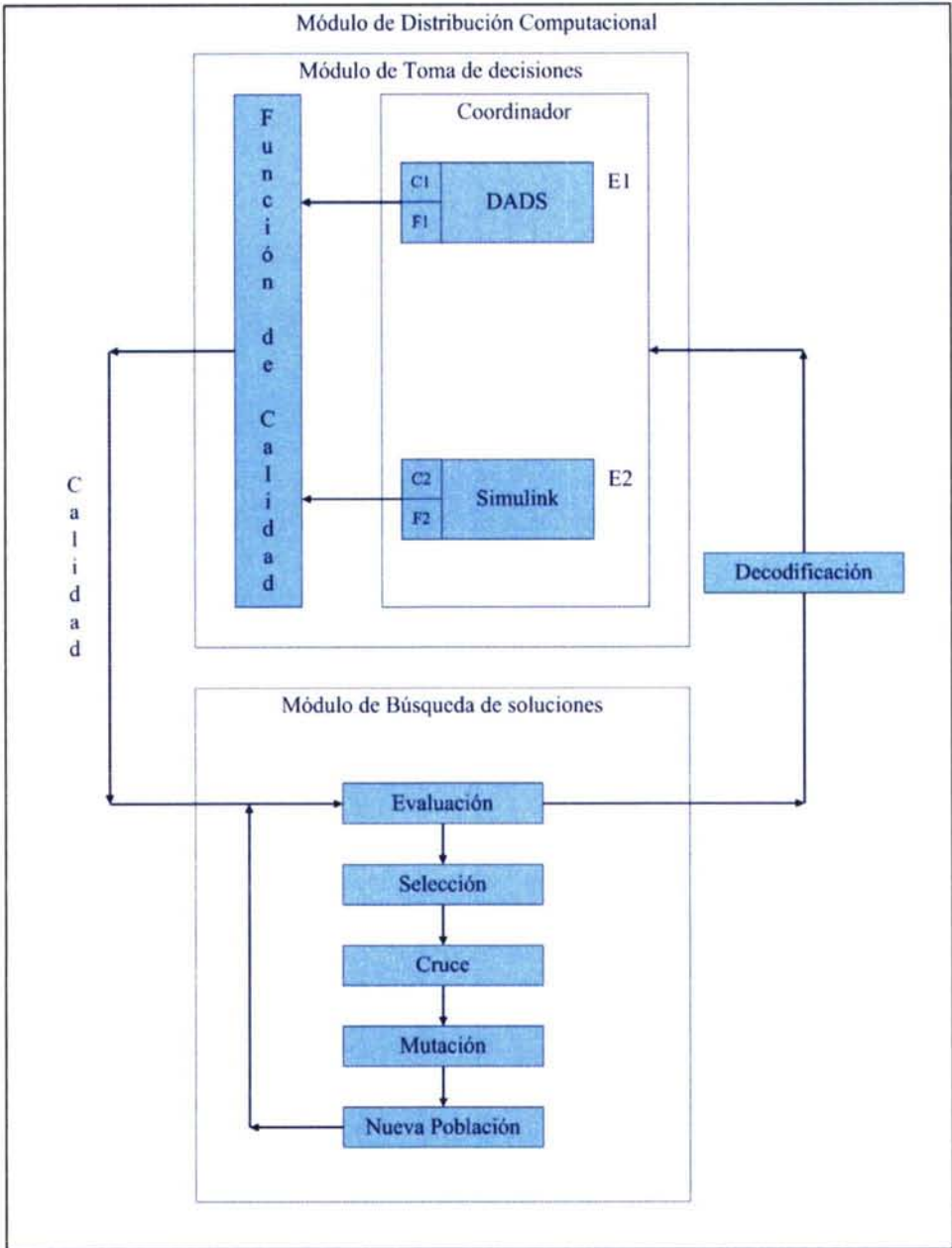


Figura 6.2: Esquema del Entorno Evolutivo de Diseño Automático implementando.

El módulo de Toma de decisiones está formado por dos simuladores. Uno de ellos, el simulador dinámico comercial DADS, exporta el modelo tridimensional del AUV generando las ecuaciones diferenciales que describen la dinámica del vehículo. El segundo simulador, Simulink de Mathworks (Figura 6.3), sintetiza el controlador creado a partir de una red neuronal cuyos pesos son descritos por el fenotipo que entra al módulo y calcula los parámetros de salida (sensores del AUV). Estas salidas definen la función de calidad que evalúa la aptitud de cada controlador. Un coordinador se encarga de integrar ambos simuladores.

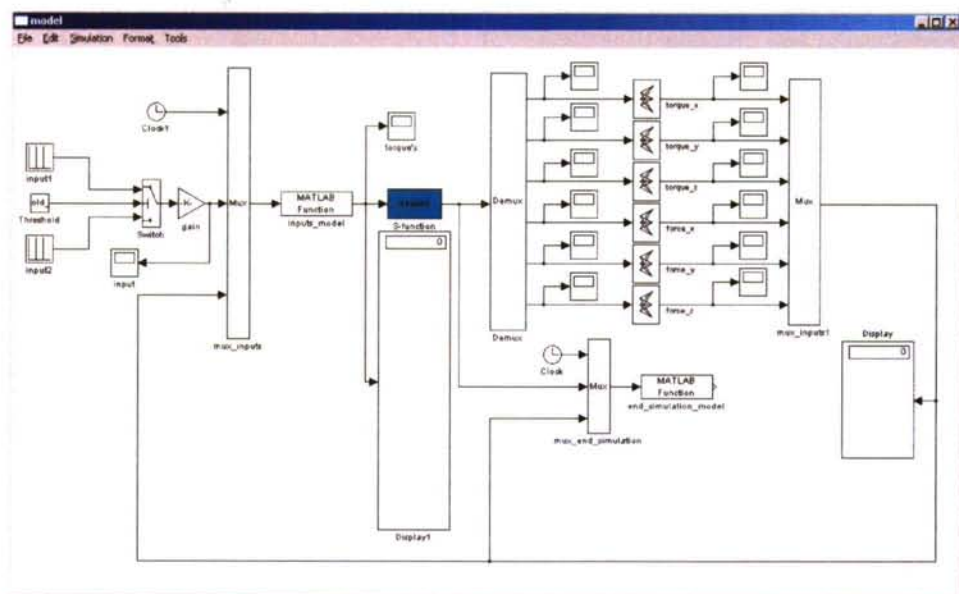


Figura 6.3: Esquema en Simulink de los controladores de estabilización de un AUV (modelo creado por DADS).

6.4.1.1 Hovering sin brazo excavador

El comportamiento dinámico del AUV con y sin manipulador fue simulado en el simulador dinámico DADS de CADSI.

El proceso para la obtención de los controladores de estabilización implicó la evolución de una población de 100 RNAs candidatas (cada una con dos entradas, dos capas ocultas con dos neuronas cada una y una capa de salida que proporciona las fuerzas de propulsión).

La aptitud de cada controlador se determinó como el valor inverso de la suma de los errores obtenidos en cada simulación medidos como el tiempo necesario que necesita el controlador para estabilizar el AUV dentro de un margen muy estrecho (el 5%) ante perturbaciones aleatorias que generaron pares de torsión entre -1 y 1 N.m.

Obviamente, al principio de la evolución muchos controladores no proporcionaron ningún tipo de estabilización y, por lo tanto, su aptitud fue muy baja. Sin embargo, tras las etapas de selección y reproducción, se obtuvo una población de individuos descendientes de mayor calidad sintetizados como combinación de los padres de mayor aptitud. La selección fue de tipo torneo, con una presión selectiva muy baja para prevenir convergencia prematura, el cruce es tradicional por un punto y la mutación aleatoria de un 4 %. Finalmente, los descendientes sustituyeron a sus padres y comenzó una nueva generación.

Para evitar ruido en la evolución debido a la duración finita de cada simulación se promediaron 5 de éstas para generar la calidad de cada individuo. Por este motivo la distribución computacional del entorno de diseño se hizo imprescindible. De hecho, para este caso en concreto, se estimó que para que la evolución alcanzase resultados significativos (cerca de 40 generaciones) debían realizarse 20000 simulaciones hidromecánicas equivalente a 2 días de simulación no distribuida. Estos datos contrastan con las 2 horas de cómputo necesarias para la obtención de los resultados presentados y que se realizaron en un clúster de 28 PCs mediante MPI estándar.

La aptitud del mejor individuo de cada generación y la aptitud media de toda la población a medida que la evolución progresa se muestra en la figura 6.4. En esta figura se puede observar claramente que después de aproximadamente 25 generaciones, la aptitud del mejor individuo ha alcanzado un valor óptimo. La segunda gráfica muestra cómo la aptitud media de toda la población aumenta gradualmente, pero a un ritmo más lento que el mejor individuo. Esto indica que la variedad de la población se conserva y que se evita el problema de tener un superindividuo.

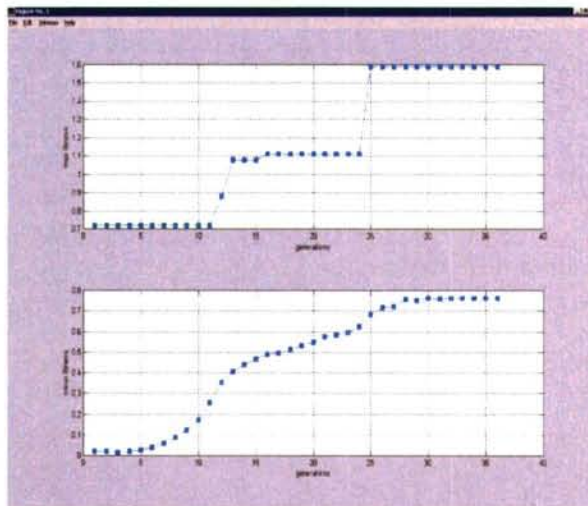


Figura 6.4: Calidad del mejor individuo (arriba) y calidad media de la población conforme avanza la evolución (abajo).

Es necesario mencionar que a pesar de que el controlador obtenido por el entorno de diseño permitió que el AUV mantuviera un comportamiento casi perfecto, existe un retraso en la respuesta de los propulsores que depende de la intensidad relativa de la perturbación con respecto a las fuerzas que los propulsores son capaces de ejercer. Además, como la aptitud fue obtenida como dependiente de los tiempos de estabilización, las respuestas muy bruscas que podrían conducir a oscilaciones en velocidad fueron evitadas por el algoritmo.

La figura 6.5 en su gráfica superior muestra la fuerza ejercida por los propulsores bajo el control neuronal obtenido y la evolución de la velocidad angular del AUV después de la aplicación de una perturbación de torsión de -1N.m ejercida durante 0.1s . Esta figura demuestra claramente cómo el controlador responde a las perturbaciones según lo esperado dentro del rango de las aplicadas durante la evolución. Lo más importante es el hecho de que al tratar con redes neuronales artificiales para la síntesis de los controladores se esperaba ver una generalización del comportamiento del AUV para casos que no le fueron presentados durante la evolución.

La gráfica central de la misma figura muestra este último caso en donde se ha desestabilizado el AUV con una perturbación relativa de torsión 2.5 veces mayor de la considerada en la evolución. Como queda claro en la figura los propulsores realizan un sobreesfuerzo para alcanzar su objetivo (obtener una velocidad cero) motivado por la pequeña fuerza máxima que pueden ejercer.

Finalmente, en la gráfica inferior se muestran las fuerzas ejercidas por los propulsores y la velocidad final del AUV como respuesta a una serie de pulsos de perturbación. Los resultados son bastante buenos, especialmente si uno considera que durante la evolución el sistema respondió solamente a perturbaciones puntuales.

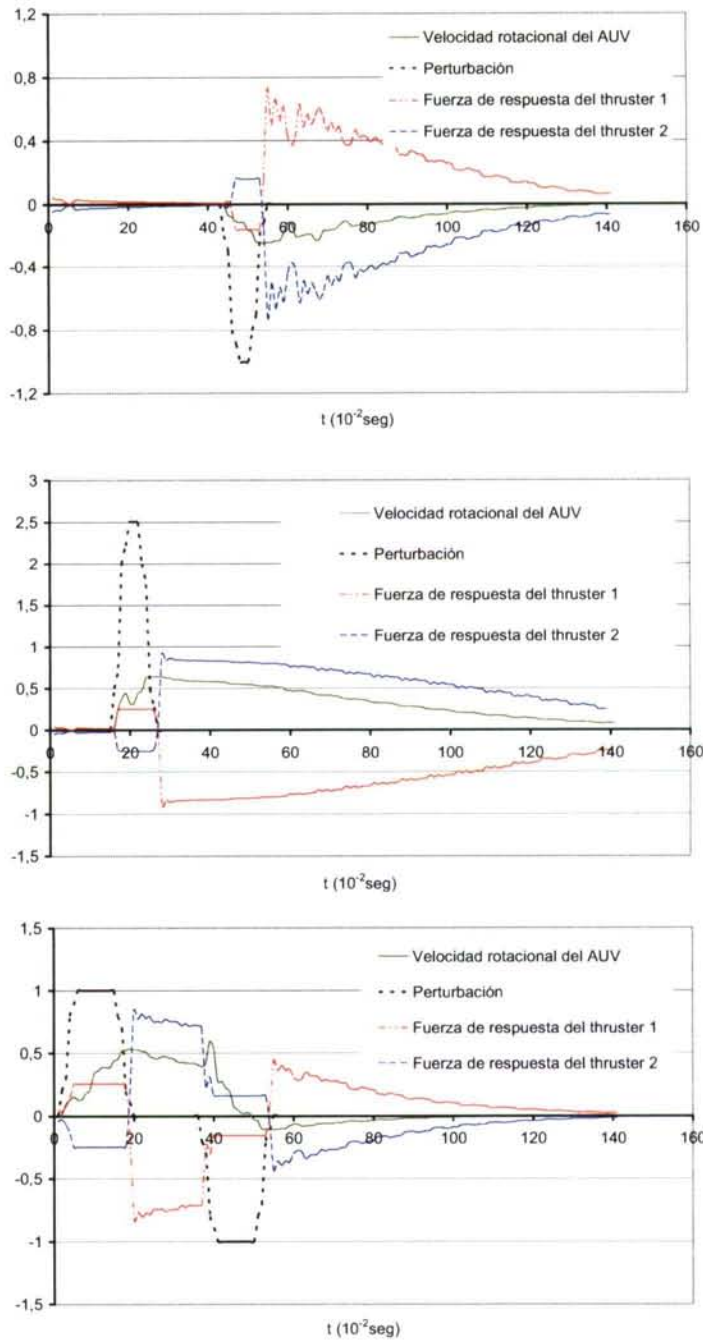


Figura 6.5 Fuerza ejercida por los propulsores y velocidad angular del AUV ante perturbaciones de torsión tipo pulso.

6.4.1.2 Hovering con brazo excavador

La gama existente de UUVs en el mercado abarca desde los que sólo permiten capturar imágenes hasta los que interactúan con su entorno incorporando un sistema de manipulación como, por ejemplo, el compuesto por un brazo mecánico. Un sistema de estas características permite ampliar enormemente las posibilidades de aplicación de los UUVs pero necesitan de una potencia elevada que se incrementa con la profundidad. En este caso aumenta también la complejidad de la plataforma en superficie que se especializa de su maniobra.

Para facilitar la operación del manipulador este tipo de sumergibles dispone de un controlador de manipulación y otro de estabilización. El primero se encarga de la manipulación del brazo (mediante el auxilio de sensores de posición) y de la supervisión en todo instante de la carga máxima permitida por la estructura del brazo mecánico. Esto último asegura que ni la muestra ni el manipulador sufran daño alguno. El segundo proporciona cierta estabilización evitando el efecto de las perturbaciones existentes en el entorno.

Este tipo de manipuladores mecánicos son ampliamente utilizados en muchos ROVs pero no ocurre lo mismo para la mayoría de los AUVs. Esto es así porque para un robot de grandes dimensiones, los efectos del movimiento de un manipulador en el cuerpo principal pueden ser insignificantes y el cuerpo principal y el manipulador se pueden considerar como dos sistemas separados. Sin embargo, para un robot pequeño los efectos acoplados del cuerpo principal y el manipulador son significativos y se deben considerar en el diseño total del sistema de control. Con el manipulador unido al vehículo, el sistema total se convierte en un sistema multi-cuerpo.

El cuerpo principal del vehículo se mueve continuamente en el agua y el alto rendimiento del control del manipulador en términos de la velocidad y de exactitud, requiere la información de gran exactitud sobre la posición y la velocidad del vehículo. La mayoría de los sensores comerciales para la posición y la velocidad del vehículo no resuelven los requisitos de exactitud del control de manipuladores. Por lo tanto, hay muchos problemas de la ingeniería para los vehículos con manipuladores.

Se han realizado pocos trabajos de investigación respecto al movimiento coordinado de vehículos submarinos acoplados a un manipulador. Todos ellos concluyen que la interacción dinámica entre el manipulador y el vehículo puede ser de gran relevancia. La existencia de una estrategia de control para el movimiento coordinado del sistema en su conjunto junto con un modelo exacto de las fuerzas hidrodinámicas de interacción es crucial para el mantenimiento en posición del vehículo y el manipulador.

- Mahesh et al. han desarrollado un control, usando una aproximación discreta del modelo dinámico para controlar el vehículo y el manipulador simultáneamente [Mahesh, 91].
- McLain et al. han realizado experimentos en el Monterey Bay Aquarium Research Institute (MBARI) que usaba el vehículo OTTER (nutria) que demostraban la dificultad de la síntesis de un controlador de este tipo [McLain, 96].
- Shoults ha investigado un esquema de control no lineal simultáneo de la posición y la orientación del sistema vehículo-manipulador [Shoults, 96].
- Canudas-de-Wit et al. han diseñado un control no lineal robusto para la compensación de los efectos del acoplamiento [Canudas-de-Wit, 98].

- Antonelli y Chiaverini propusieron un control cinemático del sistema [Antonelli, 98].

La conclusión general de estos estudios reflejan que el efecto de la hidrodinámica de cada articulación del manipulador [Mahesh, 91], [McMillan, 95] y el efecto de la dinámica de los thrusters [Yoerger, 90] ha de ser considerado en el modelado del vehículo y especialmente a velocidades lentas.

Teniendo en cuenta esto último en este trabajo se modeló el vehículo con manipulador con minuciosidad y se implementó el controlador de estabilización para que se mantuviera en la misma posición a pesar de las reacciones y perturbaciones que sobre él pudieran aparecer.

En este caso, el conjunto de las redes neuronales que realizan el control sobre cada uno de los impulsores consiste en cuatro capas de neuronas. La capa de entrada presenta tres entradas (velocidad, posición y aceleración), las capas ocultas son dos con tres neuronas cada una y la capa de salida presenta una única neurona que proporciona la fuerza ejercida por los propulsores.

El metodología evolutiva codificó cada red neuronal mediante una representación genotípica directa de 21 pesos. Cada proceso de evolución partió de una población aleatoria de 200 individuos que evolucionó más de 50 generaciones mediante el uso de un operador de selección tipo torneo con una ventana de dos individuos (baja presión selectiva) y otro de mutación con una probabilidad del 5%.

La clave del proceso evolutivo se centró en la síntesis de la función de aptitud. En este caso se planteó como la inversa de la suma de los tiempos de estabilización del AUV para mantenerse en las proximidades del estado deseado ($\pm 5\%$). Este estado está definido por términos nulos de desplazamiento, velocidad y aceleración proporcionados por el sistema sensorial del AUV para que éste se mantenga en una determinada posición sin ser afectado por la acción de cualquier perturbación existente en su entorno incluyendo las fuerzas de reacción debidas al movimiento de su brazo manipulador.

Para la evaluación del comportamiento de estabilización del AUV frente cualquier tipo de perturbación es necesario hacer uso un simulador mecánico. Como se comentó antes, en nuestro caso esta función fue realizada por un simulador 3D (DADS) comandado por un programa de control basado en MATLAB y Simulink que implementó las RNAs, proporcionó las señales de control y perturbación e implementó del Algoritmo Genético del entorno.

Una evolución típica implicaría cinco simulaciones por individuo lo que significa para un caso muy simple 50000 simulaciones por generación. Si consideramos que el tiempo de evaluación de cada individuo se aproxima a los 2 minutos, el tiempo de evolución total asciende a dos meses y en casos más complejos puede alcanzar cientos de veces este número. Por lo tanto y por razones de velocidad y eficacia computacional el uso del entorno de diseño distribuido en un cluster de PCs se hace imprescindible y reduce el tiempo de cálculo de dos a cuatro horas para cada proceso.

Al igual que en el ejemplo anterior el proceso de evaluación para el caso que nos ocupa consistió, en concreto, en la perturbación del sistema con diversas fuerzas aleatorias, lineales y angulares en los intervalos de $\pm 5\text{N}$ y de $\pm 5\text{Nm}$. La aptitud de cada individuo fue calculada en función de su comportamiento medio en 5 simulaciones y durante 5

segundos de tiempo real (equivalente 20 segundos de simulación). Es decir, para la dirección y tenemos:

ts = tiempo de establecimiento (5%)

it = iteración

phi_y = ángulo en la dirección y

$phidy$ = velocidad en la dirección y

$phiddy$ = aceleración en la dirección y

$$calidad = \left(\sum_{it=1}^{it=3} \{ts_phi_y(it) + ts_phidy(it) + ts_phiddy(it) + ts_ty(it)\} \right)^{-1}$$

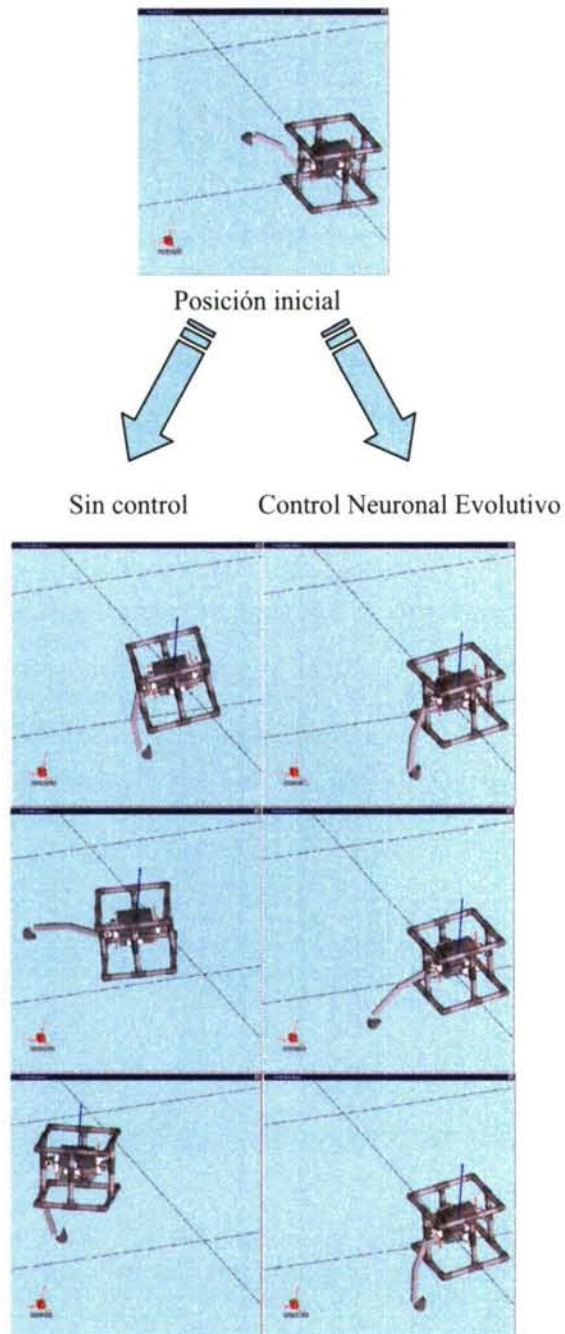


Figura 6.6 Comportamiento del AUV con y sin control mientras se mueve el brazo excavador y se aplica una fuerza perturbadora.

En la figura 6.6 se muestran los resultados obtenidos por el proceso evolutivo en dos series de imágenes donde se aprecia la diferencia entre la versión controlada e incontrolada del AUV. En la parte izquierda de la figura se puede apreciar claramente cómo después de una perturbación y sin ningún tipo de control el AUV se desplaza desde su posición y gira sobre sí mismo mientras se mueve.

Además, debido al movimiento del brazo excavador, oscila como reacción a él. En la parte derecha del figura mostramos los mismos instantes de tiempo para una simulación donde están actuando los controladores evolucionados. El resultado es que el AUV no se mueve, es estable y se compensa para la perturbación y para el movimiento del brazo.

Los gráficos de la figura 6.7 y 6.8 representan los valores de los diversos parámetros con y sin actuación para un AUV que experimente una serie de perturbaciones (cuya aceleración se muestra en las gráficas superiores). En la figura 6.7 se muestran las perturbaciones angulares y en la 6.8 las perturbaciones de desplazamiento. Del análisis de las gráficas centrales se desprende que respecto a las rotaciones, después del posicionamiento inicial del sistema en la posición objetivo, el sistema de control mantiene esta posición con una oscilación muy pequeña independientemente de las fuerzas aplicadas. Para hacer esto, el sistema de control aplica comandos de impulsión a los propulsores que da lugar a las fuerzas mostradas en la parte inferior de la figura. Con respecto al desplazamiento, está claro que el desplazamiento (mostrado en la figura en el eje z, aunque también aparece en los otros ejes) es compensado por el sistema de control.

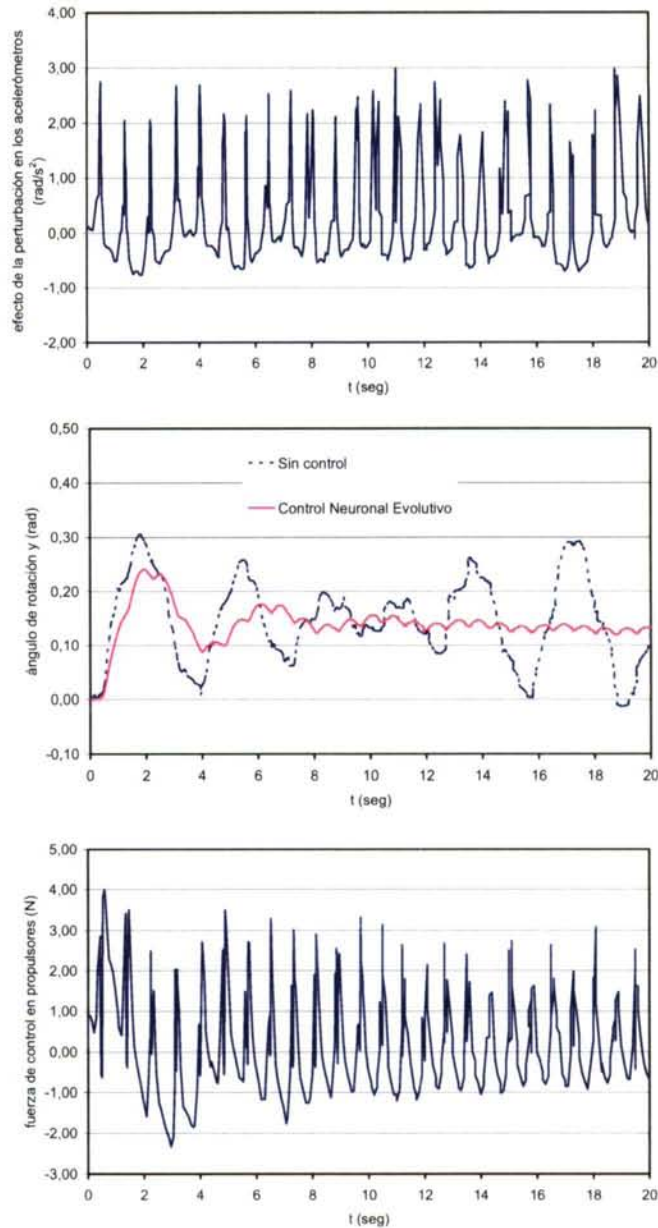


Figura 6.7 Comportamiento del AUV bajo el efecto de una perturbación tipo pulso-rotacional mientras se mueve su manipulador. La figura superior muestra el efecto de las perturbaciones, la central la respuesta con y sin control y la inferior la actuación proporcionada por la red neuronal a los propulsores.

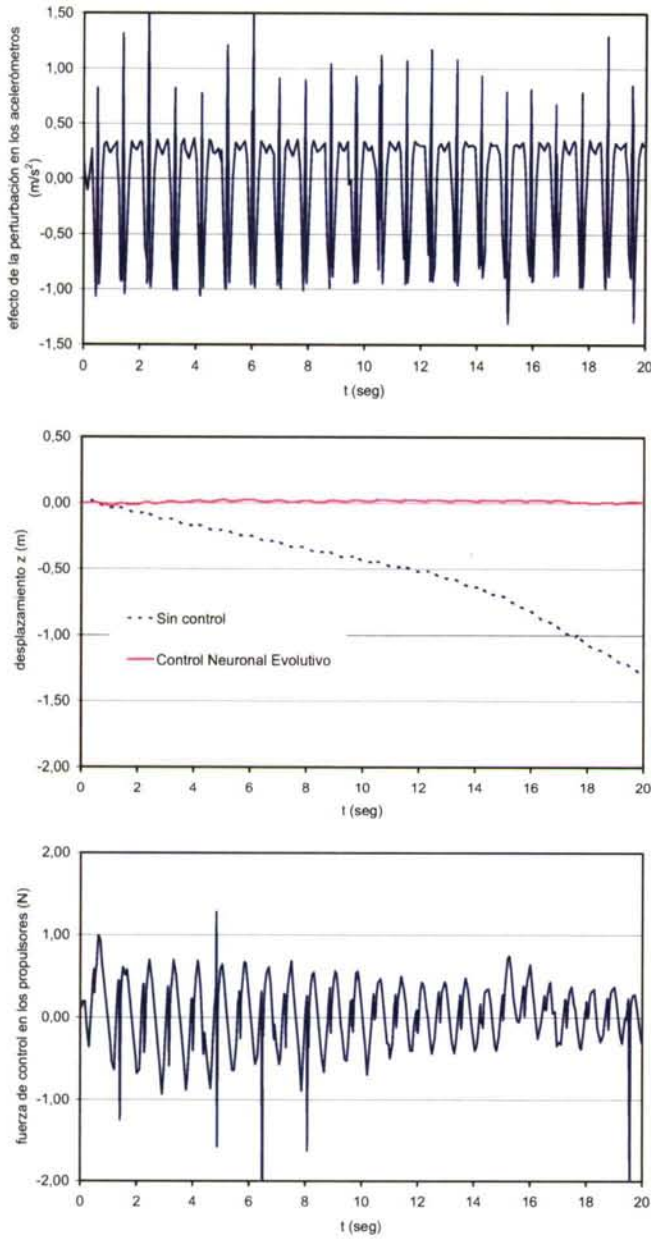


Figura 6.8 Comportamiento del AUV bajo el efecto de una perturbación de desplazamiento mientras se mueve su manipulador. La figura superior muestra el efecto de las perturbaciones, la central la respuesta con y sin control y la inferior la actuación proporcionada por la red neuronal a los propulsores.

6.4.1.3 Conclusiones del diseño de controladores para hovering

En este apartado se ha presentado una forma estructurada de diseñar controladores complejos automáticamente integrando simulaciones mecánicas e hidrodinámicas en una metodología evolutiva de la optimización. Los controladores obtenidos así son mucho más robustos que los diseñados a mano y hacen generalmente un uso más óptimo de los recursos debido a la metodología empleada para obtenerlos. Se pueden obtener controladores para otras funciones o combinaciones de funciones tan fácilmente como éstos.

6.4.2 Control de Maniobras para Catamaranes Submarinos

El trabajo de investigación realizado en este apartado consiste en el diseño automático de controladores de maniobra para un catamarán sumergible mediante el uso del entorno distribuido desarrollado en esta tesis.

Este desarrollo pertenece al proyecto de investigación PROTECAS N. 1FD1997-1262 MAR de FEDER realizado en colaboración con la empresa SUBIBOR.

Más adelante se presentarán los comportamientos de este tipo de vehículos con la actuación de los diferentes controladores neuronales sintetizados por el entorno. Los controladores actuarán sobre los timones de profundidad y dirección y la planta impulsora en función de las consignas de maniobra indicadas y procurando evitar el desequilibrio que supone la aparición de fuerzas perturbadoras.

En este caso particular los elementos a evolucionar son redes neuronales artificiales que serán utilizadas como controladores para el submarino. Las entradas de dichas redes serán los datos proporcionados por los sensores de posición, velocidad y aceleración (angular y lineal) y sus salidas los comandos de actuación que incidirán sobre los planos de control y los propulsores. Estas redes se codifican de forma directa por medio de un vector de pesos y parámetros de nodos para una estructura neuronal feedforward normal.

Para calcular la bondad del grupo de controladores que constituyen un individuo determinado es necesario someter el fenotipo resultante a diversas pruebas en el simulador hidrodinámico. Esto implica un número muy alto de simulaciones por generación y, por lo tanto, la utilización de una gran cantidad de recursos computacionales. Por esta razón la implementación del sistema sobre un entorno de computación distribuido (MPI) en un cluster de PCs supone cierta ventaja pudiendo escalar el entorno para abordar problemas muy complejos en un tiempo razonable.

Controlar la maniobra de vehículos submarinos es un problema complejo debido a sus características dinámicas, descritas por sistemas de ecuaciones altamente no lineales con incertidumbre en sus coeficientes y al hecho de que están sometidos a perturbaciones que son muy difíciles de medir. Los problemas del diseño de controladores apropiados se multiplican cuando se consideran submarinos con tripulación, ya que las características de la actuación han de mantenerse dentro de unos márgenes compatibles con la comodidad del usuario. Por lo tanto, los controladores han de cumplir varios requisitos simultáneos, por una parte han de poseer una calidad suficiente como para funcionar en un entorno cambiante que interactúa con el vehículo de forma continua, han de ser lo suficientemente simples como para poder operar en tiempo real y ,

finalmente, su operación ha de ser compatible con el hecho de transportar seres humanos. En el caso concreto que nos ocupa, el vehículo a controlar es un catamarán submarino de más de veintiséis toneladas con una planta impulsora de baja potencia y destinado a uso turístico.

A continuación se describirá brevemente las características del catamarán submarino a controlar y del problema de control que se plantea. Se describirá la configuración del método y se proporcionarán además las respuestas del sistema de control resultante en función de distintas consignas y una serie de comentarios y conclusiones.

6.4.2.1 Catamarán sumergible modelado y control

El problema, como ya se ha indicado, consiste en la obtención de un sistema de control de maniobra para un catamarán sumergible de forma automática. El catamarán a controlar está dedicado al transporte de pasajeros y presenta las siguientes características relevantes para la maniobra:

- Masa del catamarán: 26373 *Kgr*
- Empuje máximo: 9090 *Nw*, estimado a partir de la potencia máxima en inmersión (10 *Kw* producidos por los motores eléctricos)
- Propulsión en inmersión (2 motores eléctricos gemelos, hélices propulsoras)
- Potencia: 80 *Kw* diesel.
- Potencia propulsión eléctrica: 10 *Kw*.
- Velocidad en superficie: 9.5 nudos.
- Velocidad en inmersión: 2 nudos.
- Autonomía en superficie: 5.5 horas.
- Autonomía en inmersión: 8 horas.
- Profundidad máxima: 30 metros.
- Autonomía supervivencia: 72 horas.
- Distancia centro de gravedad al centro de flotación: 57 mm. Centro de flotación por encima del centro de gravedad (par restaurador).
- Momentos de inercia según ejes principales con origen en el centro de gravedad:
 - $I_{xx} = 7877.04 \text{ Kgr} \cdot \text{m}^2$
 - $I_{yy} = 153535.695 \text{ Kgr} \cdot \text{m}^2$
 - $I_{zz} = 153535.695 \text{ Kgr} \cdot \text{m}^2$

Los elementos sobre los que podremos actuar para realizar el control son los timones de profundidad y dirección así como los valores del empuje de impulsión. Para el estudio del control de este vehículo se va a partir de un modelo muy simple (figura 6.9 y 6.10) con dos superficies de control perpendiculares y a popa (figura 6.11), igual que los dirigibles, que es el presentado en la y que denominaremos modelo1. Para ambas superficies de control el coeficiente sustentación se va a suponer (C_L): 3.12.

El objetivo es obtener un conjunto de controladores que permitan el seguimiento de trayectorias que impliquen cabeceos, guiñadas, ascensos y descensos así como cambios de velocidad cuyas características no resulten incómodas para los tripulantes. Esto se concreta en que no se desean ángulos de cabeceo mayores de 15° ni aceleraciones superiores a 1 m/s^2 .

6.4.2.2 Geometría

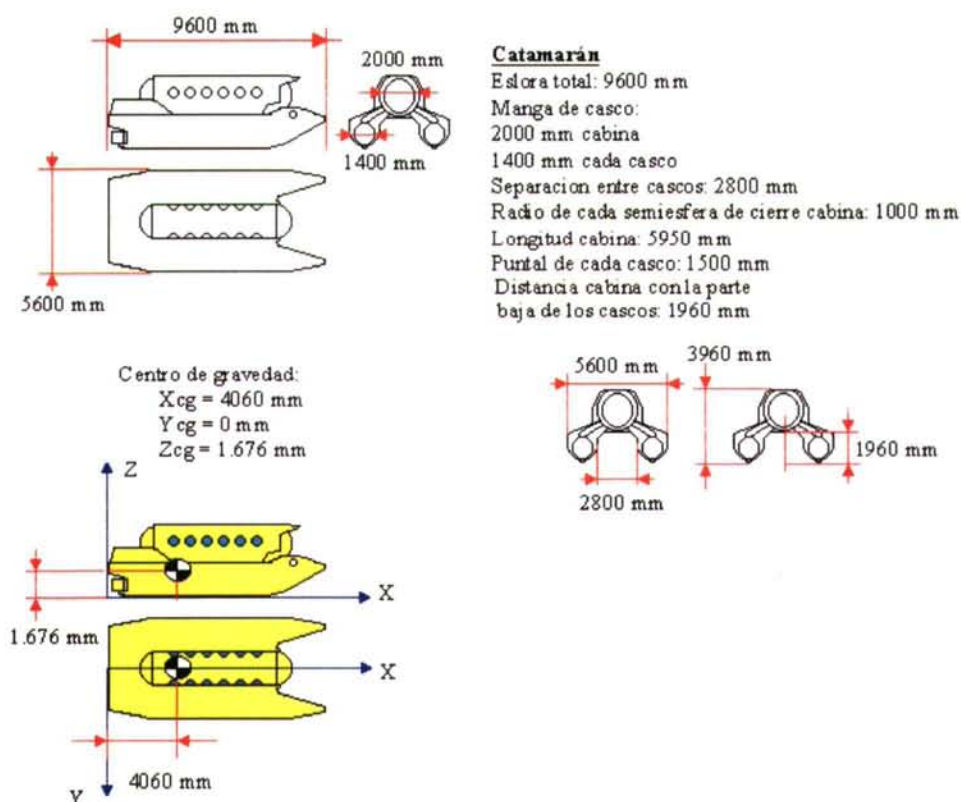


Figura 6.9: Geometría del modelo1

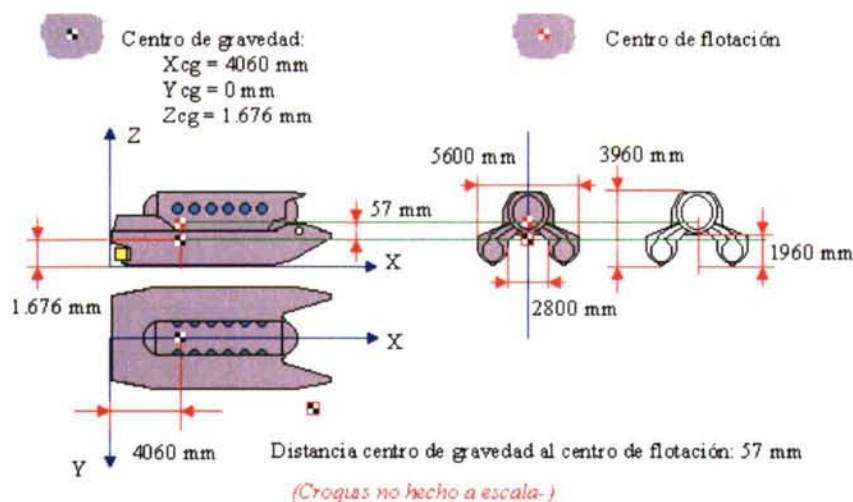


Figura 6.10: Centro de gravedad y flotación del modelo.

6.4.2.3 Superficies de control

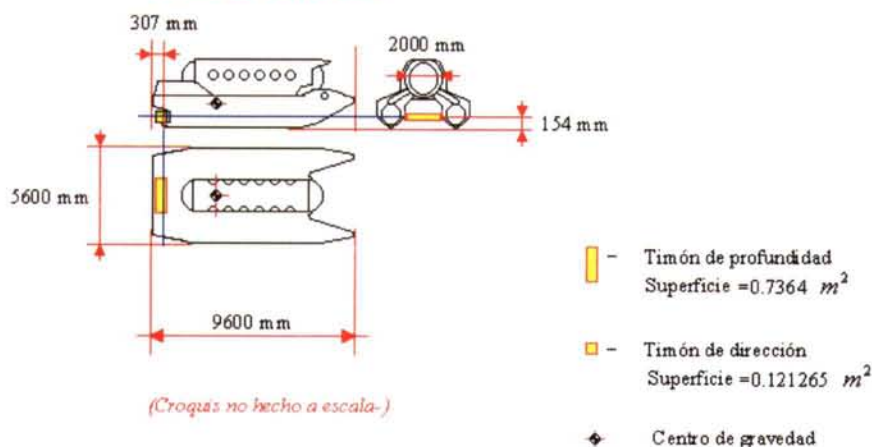
modelo 1 *Catamarán con un timón de profundidad a popa como los dirigibles*

Figura 6.11: Superficies de control.

Con el modelo 1, el modelo más simple, se hará el análisis del comportamiento dinámico por los efectos del fuselaje superpuestos a los de los timones. Existe también una configuración intermedia o modelo 2 en la que se dispone sólo de un timón de profundidad situado más a proa a 307 mm de distancia y un modelo 3 o modelo definitivo que presenta otro timón de profundidad a proa para mejorar el control. Se comprobará que los controladores sintetizados para el modelo 1 son válidos para los demás. En la figura 6.12 se muestran ambos modelos.

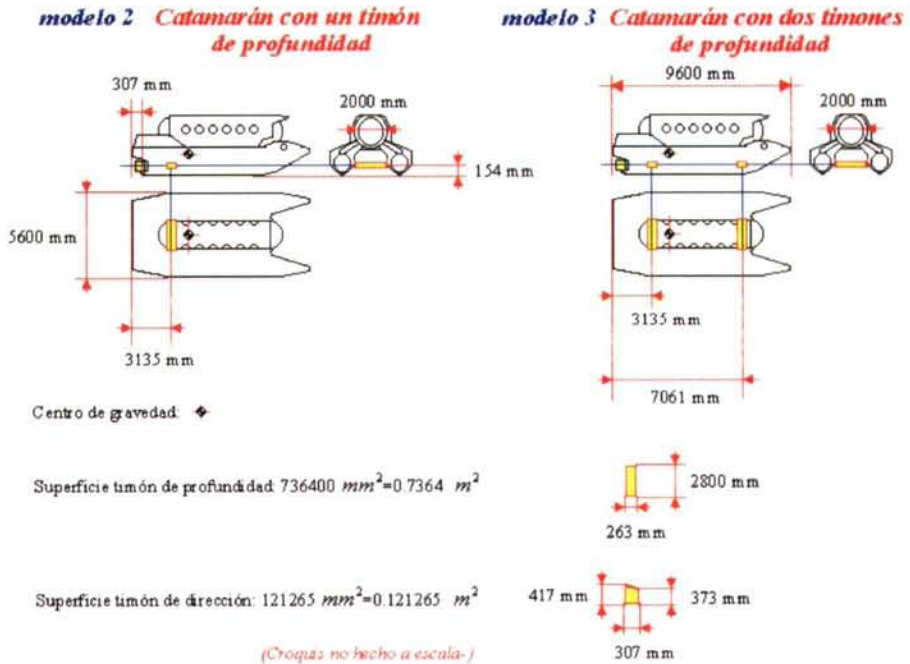


Figura 6.12: modelo 2 y modelo 3.

6.4.2.4 Criterio de signos para la deflexión de las superficies de control

Es este capítulo vamos a definir el criterio de signos empleado para la deflexión de los timones de dirección y profundidad y los ángulos de asiento, balance y guiñada.

Para esta explicación se va a emplear el modelo 1 con timones de profundidad y dirección ambos a popa del vehículo. Para el modelo 2 (el timón de dirección a popa y el de profundidad más a proa que en el modelo 1) y el modelo 3 (mismos timones que el modelo 2 con un timón adicional de profundidad más hacia la proa) este criterio es exactamente el mismo.

La combinación de los momentos como superposición de los timones y el fuselaje es extrapolable a los modelos 2 y 3. En el programa de simulación se contemplan estas particularidades.

- Criterio de signos para los ángulos de ataque y cabeceo.

En la siguiente figura se muestra la notación de signos aquí empleada.

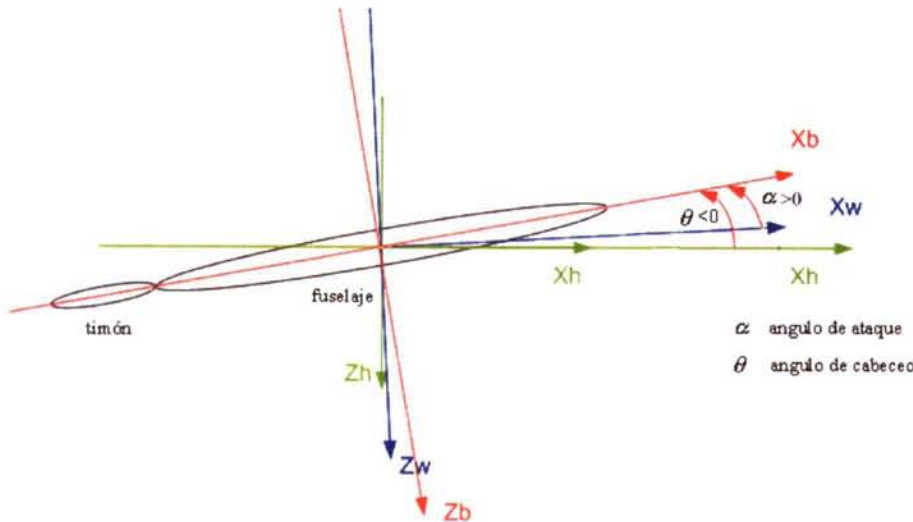


Figura 6.13: Criterio de signos en cabeceo.

- Criterio de signos en el timón de profundidad para la maniobra de cabeceo

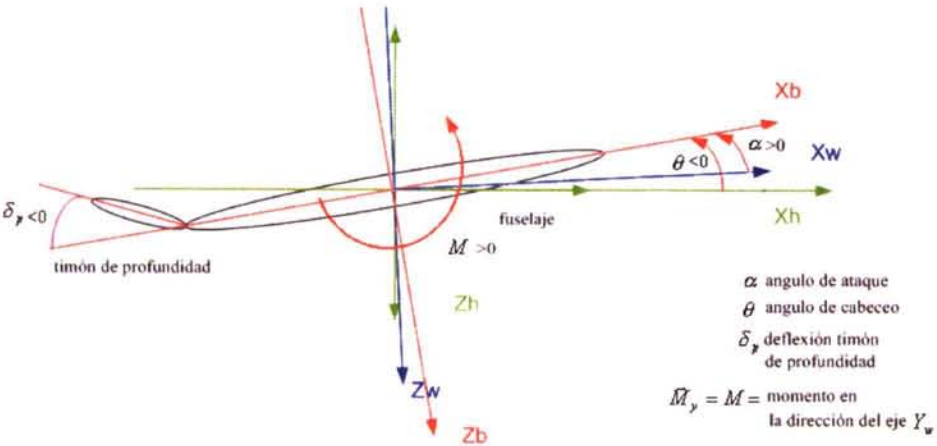


Figura 6.14: Ángulos en el timón de profundidad.

El ángulo de cabeceo θ es negativo según asciende la proa del vehículo con respecto a los ejes horizonte local.

- Criterio de signos para los ángulos de guiñada.

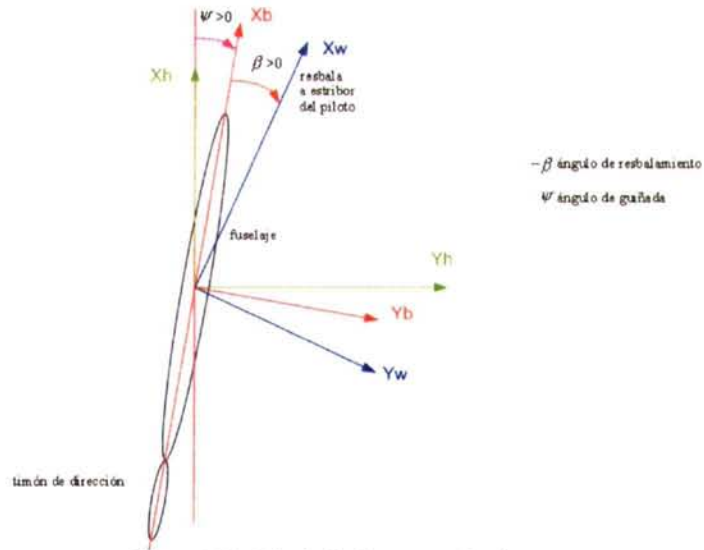


Figura 6.15: Criterio de signos en guiñada.

- Criterio de signos en el timón de dirección para la maniobra de guiñada.

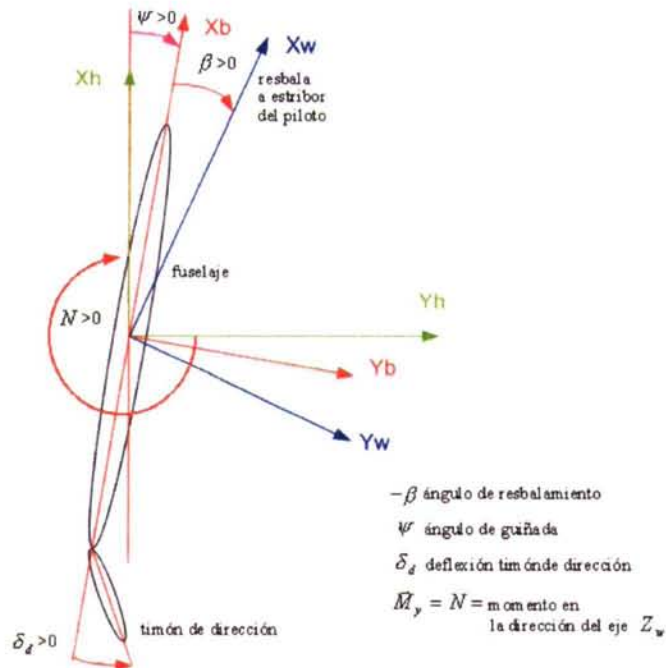


Figura 6.16: Ángulos en el timón de dirección.

De este modo quedan definidos los momentos, giros y deflexiones de los timones de dirección y profundidad. Definidos estos criterios de signos se observa que la combinación de los valores del ángulo de ataque y el ángulo de deflexión para el timón profundidad y el ángulo de resbalamiento y el timón de dirección dan distintos resultados.

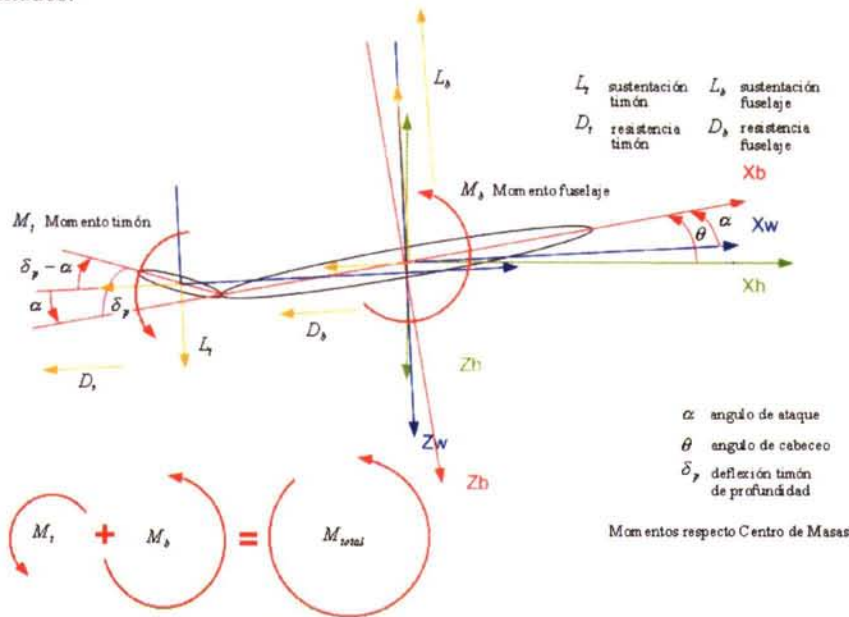


Figura 6.17: Momento del timón de profundidad positivo.

De este modo en la maniobra de cabeceo, cuando tenemos una deflexión en profundidad negativa y de valor absoluto mayor a la del ángulo de ataque se produce invariablemente un momento de cabeceo positivo como se muestra en la figura 6.5

Cuando la deflexión del timón de profundidad, en valor absoluto, coincide con el ángulo de ataque el momento del timón es nulo, considerando que el momento aerodinámico propio del timón como perfil es despreciable. Así, la única contribución al momento es la del fuselaje (figura 6.16).

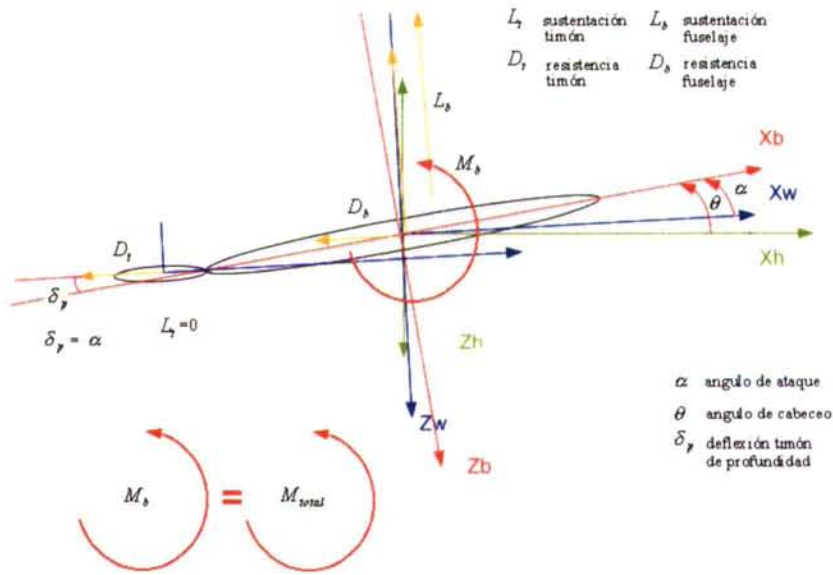


Figura 6.18: Momento del timón de profundidad nulo.

A partir de este valor conforme el valor del timón de profundidad aumenta desde valores negativos hasta cualquier valor positivo, la contribución del timón de profundidad al momento total será negativa. Incluso para valores negativos de deflexión del timón de profundidad, la contribución del timón al momento puede ser negativa.

Con valores positivos del ángulo de deflexión del timón, la contribución del timón al momento de cabeceo es mucho mayor y negativa.

Según aumentamos el valor de la deflexión del timón de profundidad, esta contribución negativa de timón va aumentando en valor absoluto hasta que anula el momento del fuselaje dando un momento total nulo. Si se sigue esta deflexión el momento total de cabeceo del sistema toma valor negativo como se muestra en la figura 6.17.

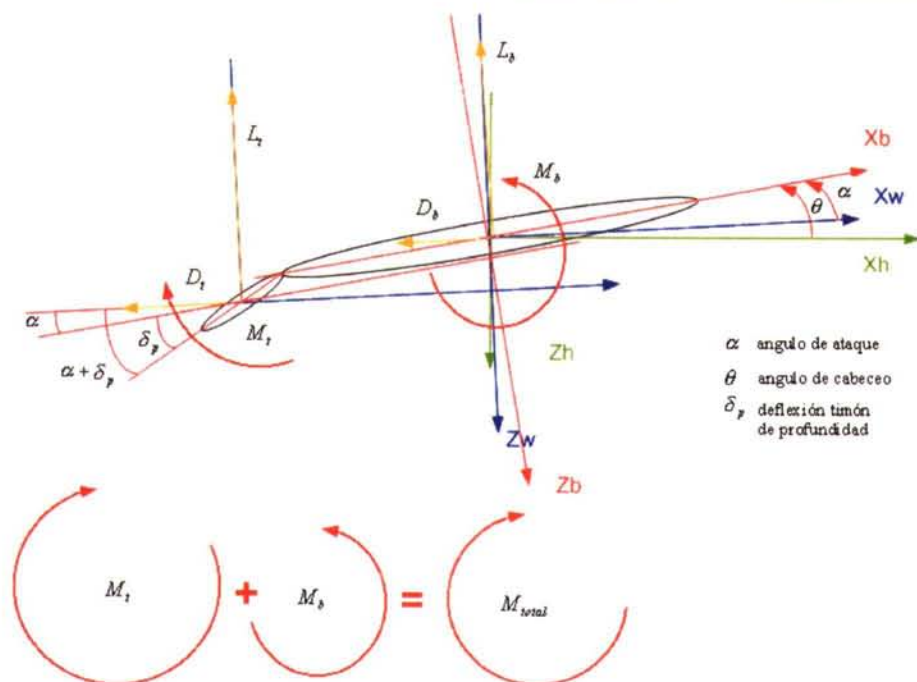


Figura 6.19: Momento del timón de profundidad negativo.

Podría parecer fácil el control en cabeceo del vehículo pero nada más lejos. Se puede lograr una posición intermedia de equilibrio (momento de cabeceo nulo) pero esa posición es muy inestable. Hay, como se ha demostrado antes de una forma simplista, un punto de equilibrio en el que no aparecería momento de cabeceo, incluso el momento negativo favorecería la estabilidad como se adelantó en páginas anteriores. Sin embargo hay unas fuerzas sobre el catamarán, como el peso, la flotación, el empuje del vehículo.. que cambian continuamente la dirección de la velocidad incidente, tanto en dirección como en módulo. En una palabra, cambian la orientación de los ejes viento, es decir, el ángulo de ataque y, por lo tanto, la configuración de fuerzas y momentos aerodinámicos sobre el cuerpo.

A modo de ejemplo, en una ascensión del vehículo, flotación mayor que peso, la velocidad incidente estará en la parte superior del vehículo por el propio proceso de ascensión. Esa incidencia hará que el cuerpo del catamarán experimente momentos de cabeceo negativos. Sólo un valor muy negativo de la deflexión del timón podrá lograr que se contrarreste ese cabeceo. Sin embargo el problema se complica ya que la dirección de la velocidad incidente dependerá del ángulo de asiento o cabeceo al ser resultado de las fuerzas másicas, peso y empuje, masa añadida y además del ángulo de ataque que regula las fuerzas aerodinámicas.

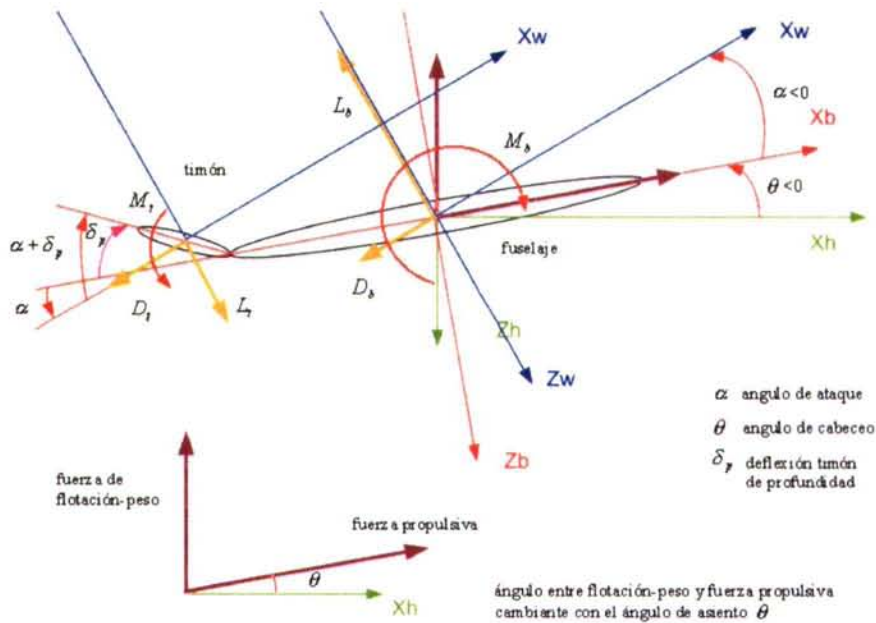


Figura 6.20: Fuerzas y momentos en el catamarán.

La resultante de fuerzas está cambiando continuamente de dirección y por tanto también lo hará la velocidad incidente con lo que implica un cambio en los valores de los momentos. Cuando empieza la ascensión, la velocidad absoluta del vehículo en el eje Z_i es nula pero conforme pasa el tiempo, las fuerzas en esa dirección, como es la flotación, hacen que la componente de esta velocidad vaya aumentando, el cuerpo se va desplazando hacia arriba, y por lo tanto, el ángulo de ataque va tomando cada vez valores más negativos. Todos los razonamientos y figuras hechos para cabeceo negativo ($\theta < 0$) son análogos a los correspondientes a cabeceo positivo.

Estas conclusiones son extrapolables a la guiñada aunque, a simple vista, la inestabilidad no sería tan grande ya que no actúan fuerzas potenciales en ese plano (plano horizonte local) que varíen continuamente con la orientación del vehículo. Sin embargo hay que tener en cuenta que tampoco actúan ningún par restaurador que da cierta estabilidad en los movimientos de balance y cabezada.

El proceso de descenso, flotación menor que el peso, es idéntico al de ascenso y las conclusiones obtenidas son casi las mismas que en el ascenso sin más que pasar la velocidad incidente a la parte inferior del plano horizonte local con ángulos de ataque positivos. Se produce una simetría completa cuando la nave es axilsimétrica, algo que no es del todo cierto ya que ni el centro de gravedad ni el de flotación cumplen esa simetría, sin embargo los razonamientos y la lógica seguida son los mismos.

6.4.2.5 Experimentos

En este apartado se presentarán una serie de pruebas de diseño automático y la calidad de los controladores resultantes.

El control se realiza por medio de la actuación de los timones de profundidad y dirección y de los motores de propulsión. Además, los distintos tipos de maniobras se especificarán por medio de consignas de asiento, velocidad, profundidad o guiñada. Se considera que la información sobre la orientación del submarino necesaria para realizar dicho control proviene de la sensorización de varios giróscopos de 3 ejes instalados en una plataforma inercial ideal, sin desviaciones. Varios acelerómetros y sus respectivos integradores dispondrán la posición del vehículo en todo instante. De este modo, obtenemos el vector de desviación de la posición y orientación del submarino con respecto a las distintas consignas (ángulos de asiento, balance y guiñada).

La complejidad del problema es palpable debido por una parte, al acoplamiento entre las ecuaciones de actuación y, por otra, a que la respuesta del sistema no es inmediata al encontrarse éste en un medio viscoso. En los lanzamientos efectuados aparecen dos grupos de variables que influyen enormemente: los ángulos de ataque y resbalamiento por un lado y, por otro, los de asiento, balance y guiñada. Según asciende o desciende el catamarán, el ángulo de ataque, los momentos y las fuerzas hidrodinámicas sobre el vehículo varían a gran velocidad. Esta variación del ángulo de ataque y la diferencia entre la dirección de los ejes viento y los ejes cuerpo hace que la dirección de avance del vehículo y la de su proa no coincida. Por otro lado, en toda maniobra de guiñada aparece un balance por la asimetría del vehículo. El único medio de control de este balance es, en nuestro caso, el par restaurador.

En los ejemplos que se presentan se estudiará la capacidad de los controladores obtenidos para responder a consignas tipo escalón y su respuesta ante perturbaciones, lo que se conoce como capacidad de mando.

En cuanto a las maniobras a realizar, se contemplan básicamente dos: giros (cabeceo y guiñada) y ascensos/descensos. Estas se pueden realizar con distintas velocidades y con diferentes ángulos de balance del sumergible. Dado el carácter de transporte de pasajeros del catamarán se considera que tanto el ángulo de cabeceo y de balance han de ser inferiores a 15° y que las aceleraciones, deceleraciones del sistema han de ser suaves.

Debido a las características físicas de los timones (no están partidos) no es posible realizar un control directo del balance sin recurrir a elementos impulsores (thrusters) y, por lo tanto, en las maniobras habrá que realizar controles que no lleven este ángulo por encima de las especificaciones, especialmente en la guiñada. Hay que tener en cuenta que cuanto más brusco el giro mayor desbalance inducido.

Es necesario que el catamarán responda a las consignas angulares de asiento y guiñada. En el momento que se especifique un ángulo determinado, el controlador ha de ser capaz de hacer que el vehículo se adapte a tal requerimiento con el menor error posible y con una respuesta en velocidad y aceleración que permita a la tripulación y a los pasajeros una travesía apacible.

En esta aplicación del entorno de diseño desarrollado en esta tesis, se ha considerado una probabilidad de mutación de un 10%, selección por torneo y una función de calidad dada por la calidad mínima de una serie de simulaciones. La valoración individual de cada simulación es el promedio de la ejecución de las consignas dadas por un conjunto de controladores durante un intervalo de tiempo determinado. En cada ejecución se valora el error cuadrático medio de la actuación respecto a consigna en todo el tiempo

En el siguiente gráfico se muestra el esquema funcional del Entorno Evolutivo de Diseño Automático desarrollado.

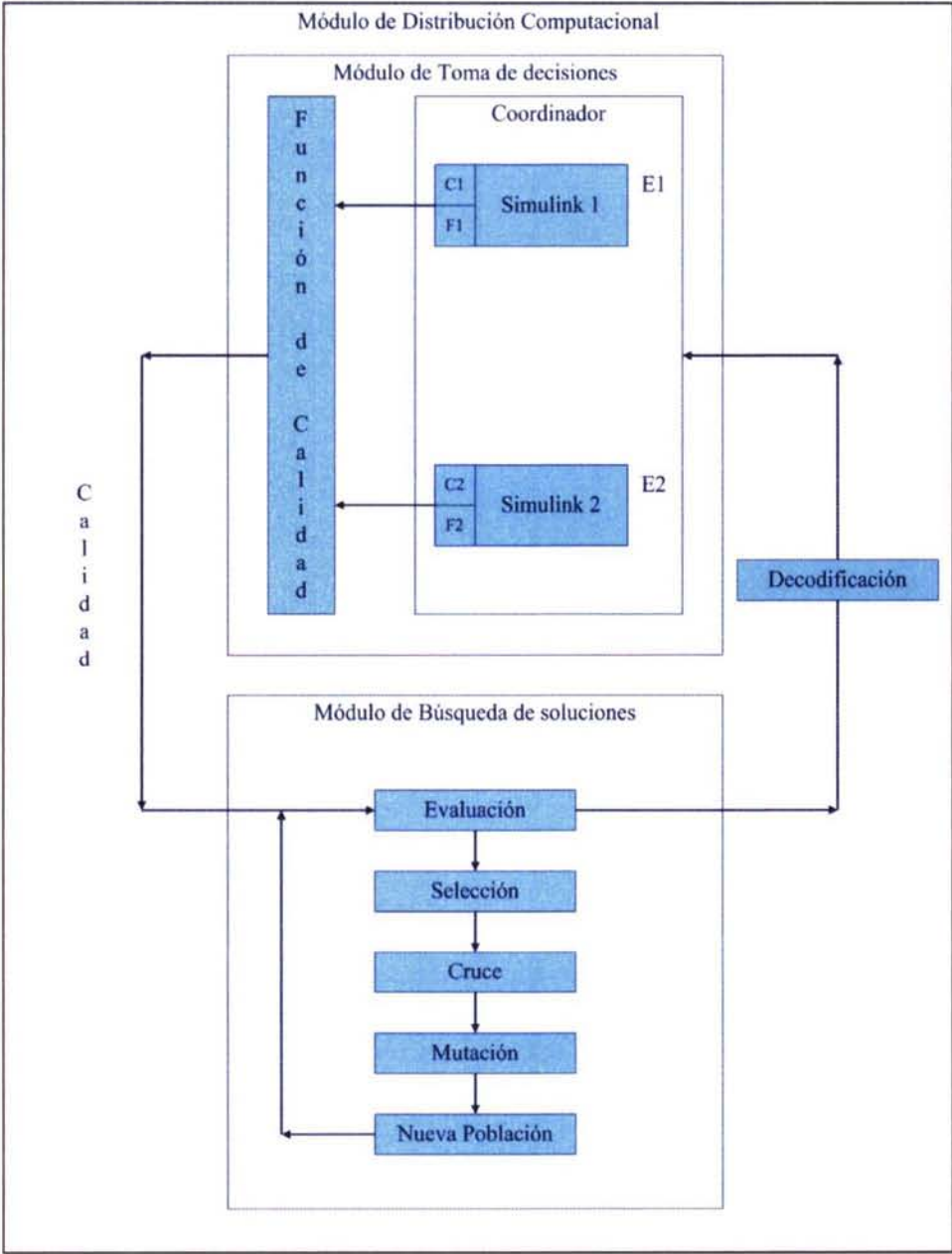


Figura 6.22: Esquema del Entorno Evolutivo de Diseño Automático.

6.4.2.6 Resultados

A continuación se presentan las gráficas correspondientes a las respuestas de los controladores obtenidos por el entorno de diseño para distintas maniobras y bajo la acción de diferentes perturbaciones. Es importante indicar que en todas las maniobras, el catamarán parte de una velocidad inicial nula y va acelerando hasta alcanzar la velocidad de consigna establecida. Inicialmente, debido a la baja velocidad, los planos de control son muy poco eficientes y, por lo tanto, grandes actuaciones producen muy poco efecto.

En principio se considera el control de cabeceo (también llamado asiento) y su respuesta ante consignas escalón bajo la acción de perturbaciones sinusoidales. Esta perturbación se simula como una corriente variable que actúa sobre el submarino en su dirección vertical con una velocidad que se indica en la figura correspondiente. En teoría este es el controlador más complicado de sintetizar debido a la gran diferencia de actuación necesaria cuando se pretende que el submarino realice maniobras de inmersión o emersión. Esta diferencia se debe a su flotabilidad.

Se evolucionó una red de 2 entradas (diferencia angular asiento-consigna y velocidad angular), 2 capas intermedias de 3 neuronas cada una y 1 salida (ángulo timón de profundidad). El proceso de evolución se realizó sobre un cluster de 50 ordenadores y los resultados para la calidad máxima y media de cada generación se presenta en la figura 6.23 donde se puede ver claramente que la máxima calidad se alcanza sobre la generación 80. Este mejor individuo es el que se ha utilizado como controlador de asiento en el submarino.

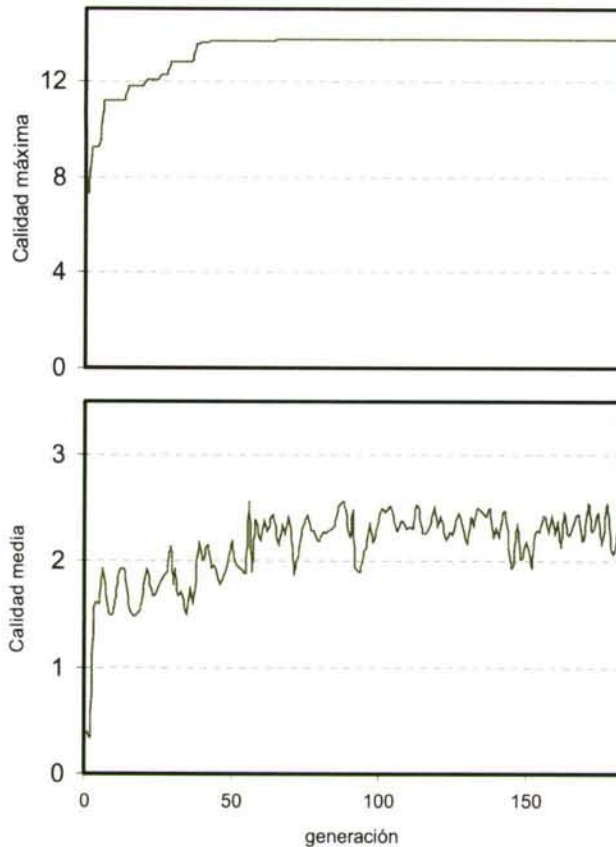


Figura 6.23: Evolución de la calidad durante la obtención del controlador de asiento. Arriba: calidad máxima. Abajo: calidad media de la población.

En la figura 6.24 mostramos la respuesta del submarino utilizando el controlador de asiento ante tres consignas extremas, asiento de $+15$, 0 y -15 grados. La línea fina presenta la respuesta del submarino en términos de ángulo de asiento frente a la consigna. La línea discontinua de esta figura representa lo que ocurriría con el submarino sin la utilización de dicho controlador (aunque sí de los demás). Como se aprecia, la respuesta es rápida pero con una oscilación lo suficientemente baja como para permitir una agradable travesía para los pasajeros. Las condiciones bajo las que se ejecutó la evolución primaban este tipo de actuación, aunque se le podría haber indicado que no se deseaba ninguna sobreoscilación a costa de una respuesta más lenta, como se verá en el controlador de guiñada.

Una vez obtenido el controlador, si el submarino se ve sometido a perturbaciones externas el controlador las compensa. Esto se puede observar en la figura 6.25, donde se presenta la respuesta del submarino y la actuación realizada sobre el timón de profundidad cuando sobre el submarino actúa una perturbación.

El comportamiento del submarino se ajusta perfectamente a los requerimientos de movimiento deseados. Inicialmente, en la primera consigna, la sobreoscilación es mayor, pero hay que tener en cuenta que el submarino parte de una velocidad nula y la actuación de los planos de control sobre el movimiento del submarino se complica. Según aumenta la velocidad, se incrementa la controlabilidad del submarino y, por lo tanto, el seguimiento de consigna mejora sensiblemente. El efecto que las perturbaciones producen en el movimiento del sistema se ve suavemente compensado por el timón de profundidad.

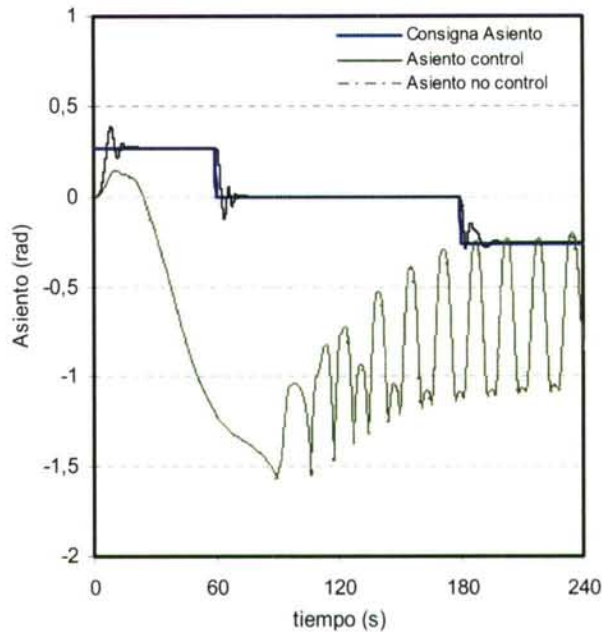


Figura 6.24: Comportamiento angular del asiento del submarino ante distintas consignas. La línea discontinua indica el comportamiento del submarino sin control

En el caso de la guiñada, como ya se ha indicado, la estrategia consistió en obtener un ángulo de guiñada de consigna respetando la limitación de que el ángulo de balanceo no debe de sobrepasar los 15 grados durante la maniobra. Se evolucionó una red de 2 entradas (diferencia angular guiñada-consigna y velocidad angular), 2 capas intermedias de 3 neuronas cada una y 1 salida (ángulo timón de dirección). Al cabo de 75 generaciones se obtuvo un controlador óptimo para este tipo de maniobra.

Para esta maniobra se presenta en la figura 6.26 el seguimiento de una serie de consignas máximas de guiñada por parte del submarino. Se puede observar perfectamente en esta figura la suavidad del seguimiento y su precisión.

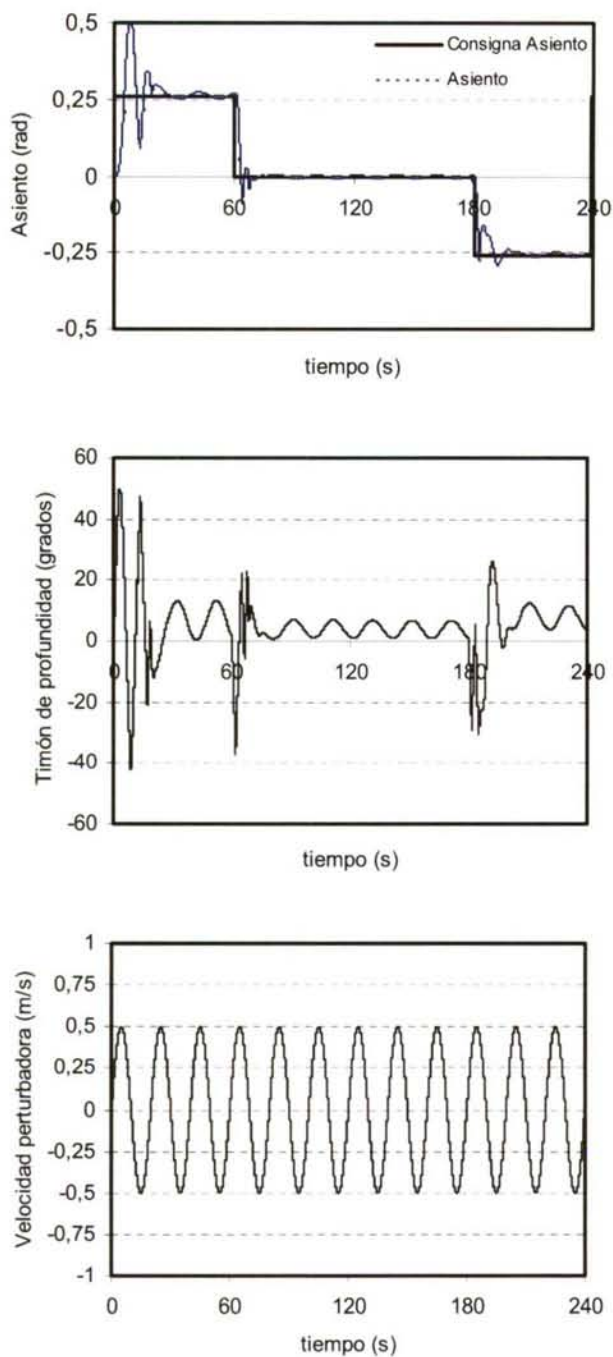


Figura 6.25: Arriba: Respuesta del submarino con el control de asiento y sufriendo corriente perturbadora. Medio: actuación del timón de profundidad. Abajo: Corriente perturbadora en el eje vertical.

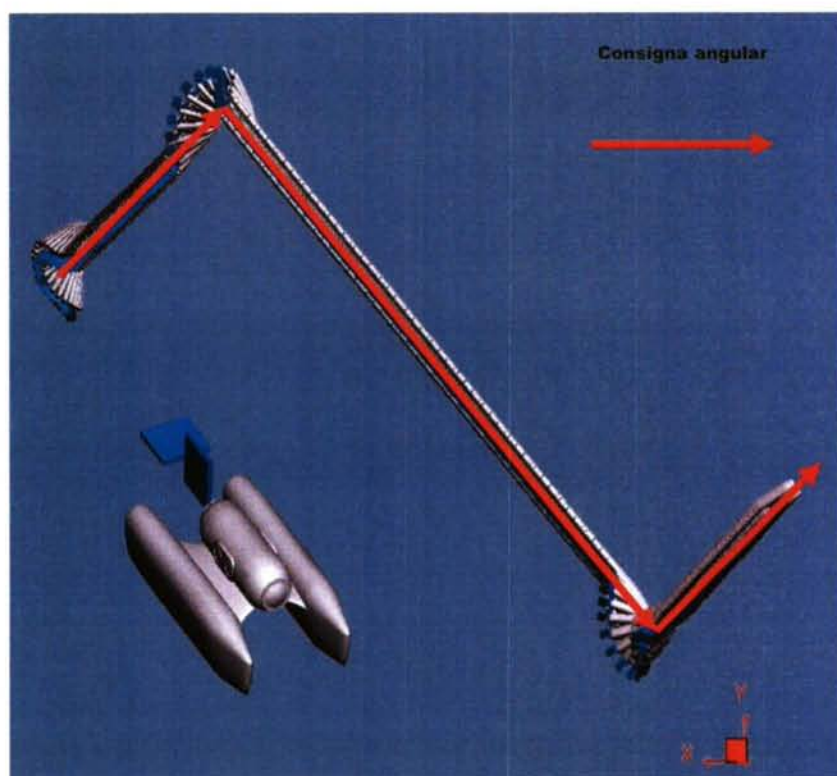


Figura 6.26: Seguimiento de consigna del catamarán considerando el controlador de guiñada.

Finalmente, en la figura 6.27 se proporcionan los mismos datos que para el asiento pero, en este caso, para el controlador de velocidad. Hay que tener en cuenta que la actuación necesaria para obtener una velocidad determinada depende del ángulo de asiento del submarino en términos de fuerzas que actúan sobre el mismo. No es lo mismo un valor nulo que un valor grande que de lugar a una emersión-inmersión brusca. En este controlador, al igual que en el de guiñada, se ha buscado una aceleración suave para respetar la comodidad de los tripulantes. Según la figura, es posible comprobar que la aceleración nunca supera los 0.7 m/s^2 , lo que resulta perfectamente cómodo.

Como ya se ha comentado, las distintas actuaciones de los controladores están acopladas. Donde se ve realmente la calidad de los resultados obtenidos, es cuando todos los controladores actúan conjuntamente. Es por esta razón por la que todas las figuras que se han presentado en esta sección correspondían a actuaciones simultáneas de todos controladores diseñados. En todos los casos el submarino ha partido de una velocidad inicial cero.

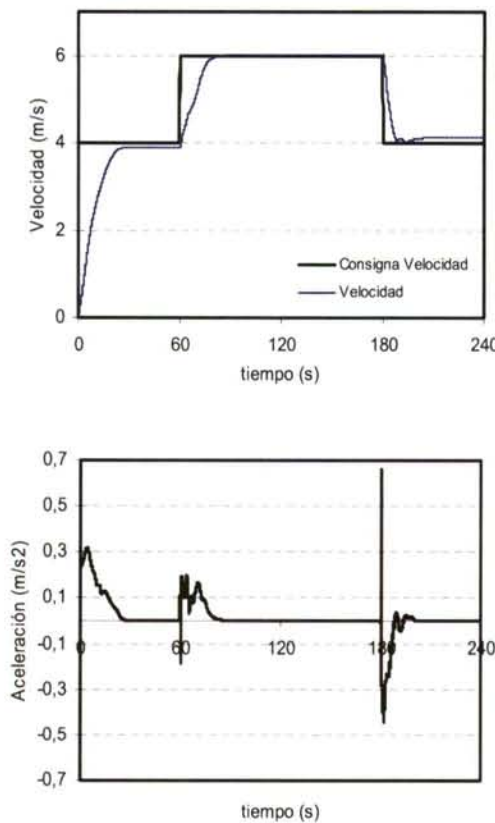


Figura 6.27: Actuación del controlador de velocidad y aceleración a la que somete a los pasajeros.

6.4.2.7 Conclusiones del diseño de controladores para maniobra

Se ha presentado la implementación del entorno de diseño para la síntesis de controladores basados en redes neuronales artificiales que controlen la maniobra de un catamarán submarino real.

Los resultados obtenidos cumplen con claridad todos los parámetros de diseño establecidos llegando a obtener comportamientos muy precisos sin recurrir a complicados modelos no lineales de la planta. El entorno permite establecer todo tipo de restricciones en la actuación que deseemos. En este caso se buscó la comodidad de los ocupantes del vehículo limitando los ángulos y las aceleraciones permitidas durante el movimiento del submarino.

6.4.3 Conclusiones del diseño de controladores

En síntesis, se han diseñado dos tipos distintos de controladores basados en redes neuronales para la estabilización y maniobra de vehículos sumergibles cuya maniobrabilidad se basaba, en un caso en thrusters y en el otro superficies de control. La dinámica del medio y el sistema y la dificultad de modelado de las interacciones

existentes da lugar a la necesidad de sintetizar controladores no lineales acoplados de gran complejidad.

El primer tipo de controladores permite la estabilización en posición de un vehículo sumergible autónomo con brazo manipulador ante la presencia de cualquier fuerza ejercida sobre el sistema incluyendo las perturbaciones externas y las reacciones provocadas por el movimiento del brazo. Los controladores no lineales generados por el entorno de diseño actúan sobre los propulsores del vehículo en función de las señales que reciben desde un sistema de detección de tal modo que la posición del submarino permanece inalterable. Se trataron dos casos en este diseño preliminar, con y sin brazo excavador, introduciendo así el complejo problema dinámico de dos cuerpos acoplados. Se estudiaron los controladores resultantes y se proporcionaron las respuestas del sistema ante distintos tipos de perturbaciones para maniobras de hovering con y sin excavación. Los controladores resultaron apropiados cumpliendo las especificaciones de diseño y permitieron en un desarrollo posterior establecer el diseño del sistema de control de un AUV real para exploración.

El segundo tipo de controladores se diseñó para la maniobra de un catamarán sumergible que compensase la acción de las fuerzas perturbadoras que actuasen sobre él mediante su actuación sobre un sistema de propulsión y una serie de timones de profundidad y dirección y en función de los datos proporcionados por sus sensores. Este sistema de control pretendía adaptarse a las consignas de maniobra de un catamarán submarino para el transporte turístico de 20 toneladas y cuya respuesta debía ser compatible con los parámetros de seguridad y confort adecuados a su función.

En este problema destaca la dificultad de la realización de cada controlador individual pues debe tener en cuenta todas las posibles perturbaciones existentes en el medio submarino y la dinámica especial de estos vehículos que da lugar a respuestas que cambian de manera extrema dependiendo de la velocidad, la maniobra que se esté realizando, la orientación del vehículo, etc. Además el funcionamiento global de los vehículos depende de la acción coordinada de varios controladores con respuestas acopladas entre sí debido a la dinámica no lineal del vehículo.

Para la evaluación de los distintos diseños obtenidos se utilizó un módulo de simulación, realizado ad hoc por otro miembro del grupo de investigación, que permitió obtener la respuesta del catamarán en cualquier circunstancia. La ejecución del entorno de diseño se realizó de forma distribuida en un cluster de 50 ordenadores.

Los resultados cumplieron todas las especificaciones, encontrándose incluso que permitían evitar problemas de balanceo por medio de una adecuada compensación de las velocidades y los ángulos de los timones de profundidad y dirección. Este resultado fue importante ya que el catamarán no disponía de ninguna forma de realizar esta compensación directamente (salvo la fuerza restauradora debida a las posiciones de su centro de masas y su centro de flotación, fuerza cuya actuación es muy lenta).

Plantas Industriales

7 Plantas industriales

En este capítulo se aplicará la capacidad de la metodología evolutiva de diseño automático desarrollada en esta tesis para la optimización de plantas industriales y su aplicación en un caso real.

Se verá como, en este tipo de problemas de gran complejidad, la relación entre los parámetros que definen la calidad del sistema de producción (productividad, rentabilidad, atascos en la línea, etc.) y los correspondientes a los subsistemas que lo conforman (controladores para la gestión de la entrada y salida de entidades) es altamente no lineal e indirecta y que la aplicación de la metodología del entorno de diseño que se propone supone ciertas ventajas.

En este capítulo se abordarán diseños desde el punto de vista morfológico y de control. Con respecto a la morfología, se estudiará la disposición óptima en planta de maquinaria y en relación al control se desarrollará un controlador secuencial óptimo para la maximización del rendimiento del proceso.

7.1 Marco teórico. Simulación de Procesos

A pesar de los avances tecnológicos, muchas empresas sufren todavía de equipos y métodos obsoletos, ineficaces y sin automatización. Esto se debe en parte al gran y costoso tiempo requerido para la exploración de métodos de trabajo alternativos y la búsqueda de nuevas tecnologías en sistemas y procesos reales. Para conseguir sus objetivos, las empresas necesitan desarrollar una posición competitiva extremadamente fuerte. Una forma de conseguirlo es mejorando los sistemas y procesos y la metodología de diseño automática desarrollada en esta tesis proporciona una herramienta para la síntesis de las optimizaciones propuestas. La versatilidad de la herramienta es tal debido a su modularidad que la simple inclusión en la misma de un simulador de procesos discretos proporciona todo lo necesario para el cumplimiento de los objetivos.

7.1.1 Sistemas vs. Procesos

En el marco de la simulación cuando se tratan líneas de manufactura se habla de sistemas discretos. Se define sistema como el conjunto de elementos interconectados tales como entidades (recursos humanos, procedimientos, materiales, equipo, información, espacio y energía) y recursos (equipo, herramientas y personal) que interaccionan entre sí con el objetivo común de realizar un proceso. Las variables globales de un sistema reflejan alguna característica del sistema completo, independientemente del tipo o cantidad de entidades que se encuentren en él pero accesibles y modificables por éstas. En el Diseño de Sistemas el énfasis recae en los detalles de cómo, dónde y cuándo.

Por otro lado, se define proceso como el método por el cual funciona el sistema. Cada proceso consta de un conjunto de actividades relacionadas cuyo fin es obtener unas salidas, generalmente un producto o servicio, a partir de unas entradas. Las actividades tienen una duración e implican el uso de los recursos del proceso. Los procesos comerciales e industriales representan la utilización y relación de elementos y recursos accionados por eventos, tales como la recepción de materias primas, petición de un servicio, etc. Por ejemplo, en el proceso de fabricación de un producto uno de los

posibles eventos sería la recepción de las materias primas, la mano de obra y las piezas conformarían las entidades y la actividad encomendada podría ser el montaje de dichas piezas. En el Diseño de Procesos se presta especial atención al que se ejecuta.

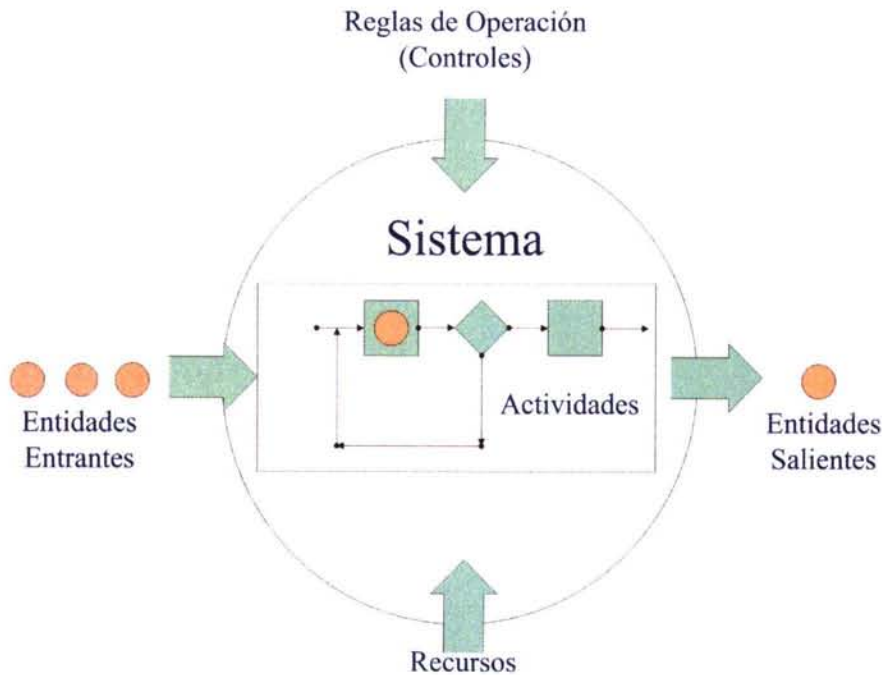


Figura 7.1: Elementos de un sistema

Las entidades son, por tanto, los elementos que procesa el sistema. Éstas son dinámicas, es decir, son creadas, cambian de estado, se mueven en el sistema durante algún tiempo afectando y siendo afectadas por otras entidades y son destruidas, cuando salen del sistema. Los productos, clientes o documentos, por ejemplo, son tipos de entidades. Cada una de las entidades procesadas en un sistema tiene sus propias características tales como costo, forma, prioridad, estado o condición. Un atributo es una característica de toda entidad cuyo valor específico la diferencia de las demás. Ej. : identificador, hora de llegada, prioridad. Las entidades se pueden clasificar en animadas (clientes, mano de obra etc.), inanimadas (equipos, pallets, etc.), o intangibles (comunicaciones, proyectos, etc.).

Un evento es algo que sucede en un instante determinado dentro del tiempo de simulación y puede hacer cambiar los atributos de las entidades. Ej. : llegada y salida de una entidad al sistema, fin de la simulación, etc.

Las actividades son las tareas o acciones que tienen lugar en el sistema, tales como satisfacer una orden de pedido, atender un paciente, reparar una máquina, etc. Las actividades tienen duración y, usualmente, envuelven el uso de recursos. Ejemplos típicos de actividades es el procesamiento de entidades (llenar un formulario, fabricación de una pieza, inspección, tratamiento, etc.), mover entidades, mover recursos y el mantenimiento y reparación de máquinas.

Los recursos son los medios por los cuales se ejecutan las actividades. Una entidad se apodera de un recurso cuando se encuentra disponible y lo libera cuando termina de utilizarlo. Un recurso puede tener una capacidad variable, que puede ser modificada durante la simulación. Definen quién o qué realiza tal actividad, dónde se realiza y cuándo se realiza. Los recursos pueden ser representados por grupo de entidades individuales llamadas unidades y pueden tener una variada gama de características tales como capacidad de proceso, velocidad, tiempo de ciclo, flexibilidad, confiabilidad, etc. Los recursos en un sistema pueden incluir personas, equipos, espacio, métodos, energía, tiempo y dinero.

Las reglas de operación o controles gobiernan el cómo, cuándo y dónde se realizan las actividades. Determinan qué acción tomar cuando ocurren ciertos eventos o condiciones. Al más alto nivel toman la forma de planes y políticas. A bajo nivel toman la forma de procedimientos o lógica de programas de ordenador.

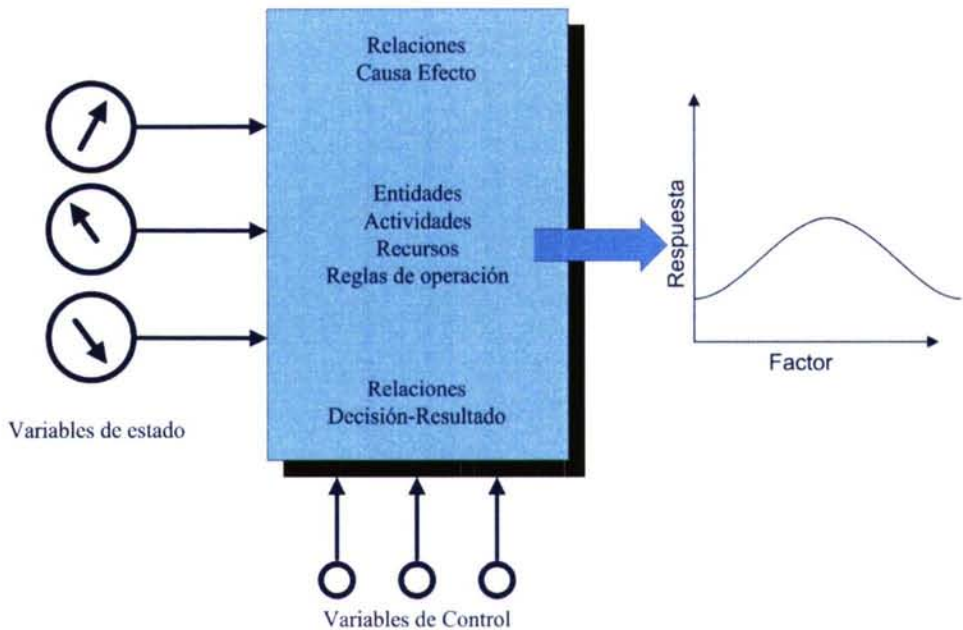


Figura 7.2: Elementos de un sistema

7.1.2 Sistemas discretos. Sistemas de Fabricación.

El objetivo común de los modelos de sistemas discretos es evaluar el rendimiento del sistema y su comportamiento para servir de guía a la optimización del sistema real.

Los sistemas discretos más frecuentes en la práctica son los de fabricación, servicio, manipulación de materiales y redes. Todos estos sistemas presentan diferencias significativas pero coinciden en que se componen de entidades individuales que fluyen por el sistema y emplean recursos. A continuación se describirán los conceptos fundamentales de los sistemas de fabricación, pues son de esta clase los sistemas que se estudiarán.

En los sistemas de fabricación se procesan materias primas y/o componentes de productos por medio de una serie de operaciones (actividades, colas y áreas de

almacenamiento) donde intervienen personal y equipos con cierto grado de automatización, hasta obtener el producto terminado. Además del producto final, se persiguen otros objetivos como bajos niveles de inventario o la satisfacción de los clientes.

Los sistemas de fabricación pueden clasificarse en tres tipos:

- Líneas de producción.

En este tipo de líneas como, por ejemplo, las líneas de montaje, los productos se procesan individualmente o en lotes en un grupo de máquinas dispuestas en serie. Los flujos de productos no se cruzan.

- Talleres.

Las máquinas del mismo tipo funcional se organizan por secciones. Los flujos de productos se cruzan.

- Sistemas de fabricación celular y flexible.

Los productos se agrupan en familias y éstas se procesan a su vez en grupos de máquinas.

La simulación de sistemas de fabricación se hace con la intención de determinar el efecto de los cambios propuestos. Así, la decisión de introducir un nuevo producto o mejorar la eficiencia implica generalmente añadir o renovar equipos, etc. Los objetivos del modelo de simulación están relacionados, en concreto, con la identificación de cuellos de botella, el cálculo de la capacidad y utilización de un sistema, el estudio del rendimiento de escenarios alternativos y el desarrollo de estrategias de programación del trabajo.

La mayor parte de las operaciones de fabricación dependen de la disponibilidad, capacidad y fiabilidad de los equipos, por eso será frecuente que los esfuerzos del modelado vayan dirigidos a resolver problemas de los equipos. Algunas consideraciones importantes que se han de tener presente al diseñar sistemas de fabricación son la cantidad y el tipo de máquinas, el tamaño y la localización de los almacenes intermedios (*buffers*), los tiempos de mantenimiento y reparación y las políticas de carga y secuenciamiento de las máquinas.

En las simulaciones tratadas en este capítulo se prestará especial atención a consideraciones relativas a máquinas y a sus políticas de carga.

Algunas medidas típicas para evaluar el rendimiento de sistemas de fabricación así como sus definiciones son:

- Tiempo de ciclo. Tiempo transcurrido desde que se piden las materias primas hasta que se entrega el producto al cliente.
- Intervalo de fabricación. Una parte del tiempo de ciclo. Es el tiempo transcurrido desde que se transforman las materias primas hasta obtener el producto terminado.
- Pendiente. Número de productos esperando a ser procesados (función del tamaño de los *buffers*).

- Utilización. Ratio del tiempo de procesamiento dividido por el tiempo disponible.
- Tasa de salida. Numero de productos fabricados por unidad de tiempo.
- Tasa de defectuosos. Porcentaje de productos defectuosos obtenidos en una operación.
- Cuello de botella. Una restricción en el flujo de producción generalmente causada por un recurso (mano de obra, etc.) con una elevada utilización.
- Tiempo de ciclo de una máquina. Tiempo medio que un equipo tarda en efectuar una operación. Suele incluir factores como tiempo de parada, fatiga del operador, etc.
- Trabajo en proceso. El nivel de inventario en proceso.
- Sobrante. Todo aquello que exceda del mínimo de recursos requerido para fabricar el producto.
- Costo acumulado. Suma de todos los costos de los recursos empleados para producir una unidad de producto terminado.

7.1.3 Diseño y Optimización de Sistemas y Procesos

El objetivo del Diseño y Optimización de Sistemas y Procesos es transformar las entradas en las salidas deseadas de la manera más eficiente en costos y tiempo requerido. El Diseño consiste en la selección apropiada de la capacidad y control (asignación de tareas y recursos), así como en el estudio de la tecnología y de los métodos necesarios para lograr los objetivos. La Optimización, por otro lado, comprende la mejora de las tareas y del flujo recursos-clientes, la minimización del efecto del mantenimiento, retardos e inventario y la gestión de calidad.

Tanto el Diseño como la Optimización de Sistemas hacen hincapié en la identificación de sus elementos y en la comprensión de las relaciones Causa-Efecto y Decisión-Respuesta.

Las relaciones Causa-Efecto definen el comportamiento o dinámica del sistema mediante la identificación de todas las acciones que pueden tener lugar en él y en la determinación de las acciones que dará origen a cada una de ellas.

Las relaciones Decisión-Respuesta vienen definidas por el valor que toman las variables de control que especifican todos y cada uno de los elementos particulares del sistema: cantidad de recursos, duración de la actividad, lógica de decisión, etc. La Respuesta o Desempeño Global es el resultado de todos los efectos combinados que ocurren en un periodo de tiempo dado. Las variables de respuesta son variables que miden el desempeño del sistema en respuesta a ciertas combinaciones de las variables de control.

Medidas típicas de desempeño de un sistema son:

- Tiempo de Ciclo. Tiempo requerido para completar el procesamiento de una entidad.
- Tiempo de Utilización de Recursos.
- Tiempo de Valor-Añadido. Cantidad de tiempo que clientes y material ocupan realmente en las operaciones o servicios.
- Tiempo de Espera. Lapso de tiempo que materiales y clientes esperan en ser atendidos por un recurso.
- Tasa de Proceso. Tasa en la cual se procesan las entidades. Mide la capacidad de procesamiento.

- Calidad. Proporción de entidades producidas o clientes atendidos que cumplen con los requisitos especificados.
- Flexibilidad. Habilidad del sistema para adaptarse a las fluctuaciones de volumen y variedad.
- Costo de operación del sistema.

En los casos prácticos que se tratarán aquí se aplicará la metodología evolutiva de diseño automático tanto para el diseño como para la optimización de sistemas.

7.2 Diseño de Morfologías y Controladores para Plantas Industriales

Partiendo de la metodología evolutiva de diseño automático desarrollada en este trabajo, se sugirió la idea de parametrizar y optimizar una línea de manufactura real. Esta metodología se basa en algoritmos genéticos para realizar procesos de optimización y necesita de la parametrización del modelo, la integración de diferentes herramientas informáticas y la automatización del proceso.

De entre las soluciones informáticas que conforman el entorno destaca el software de simulación. En este punto es necesario hacer hincapié que el software existente actualmente en el mercado para este tipo de procesos carece del suficiente carácter abierto como para poder asegurar la eficiencia de la integración con otros programas y, si ésta existe (como en el caso del software utilizado EXTEND), resulta mínima y es poco eficaz por sí misma para la realización de múltiples simulaciones. Otro aspecto a tener en cuenta es la imposibilidad de realizar simulaciones distribuidas en red sin el auxilio de otras herramientas. Esto es importante pues limita en gran medida la realización paralela de gran cantidad de simulaciones como es posible en el caso de aplicar algoritmos genéticos.

La metodología expuesta supera todos estos inconvenientes y facilita al ingeniero una herramienta de optimización potente, versátil e integrable con simuladores comerciales. En este proceso, en concreto, Visual Basic hace de enlace entre mpich (C++) y EXTEND. Desde Visual Basic se lanza el software de simulación y a partir de ahí se crearán, interpretando el cromosoma, los bloques y conexiones correspondientes. Una vez finalizada la simulación, se extraerán los resultados obtenidos de la línea desde los bloques creados para tal fin y se cerrará el programa. Los valores de aptitud que genera cada simulación corresponderán a las calidades de los individuos que constituyen una población.

El esquema funcional del Entorno Evolutivo de Diseño Automático aplicado para este caso en concreto es el siguiente:

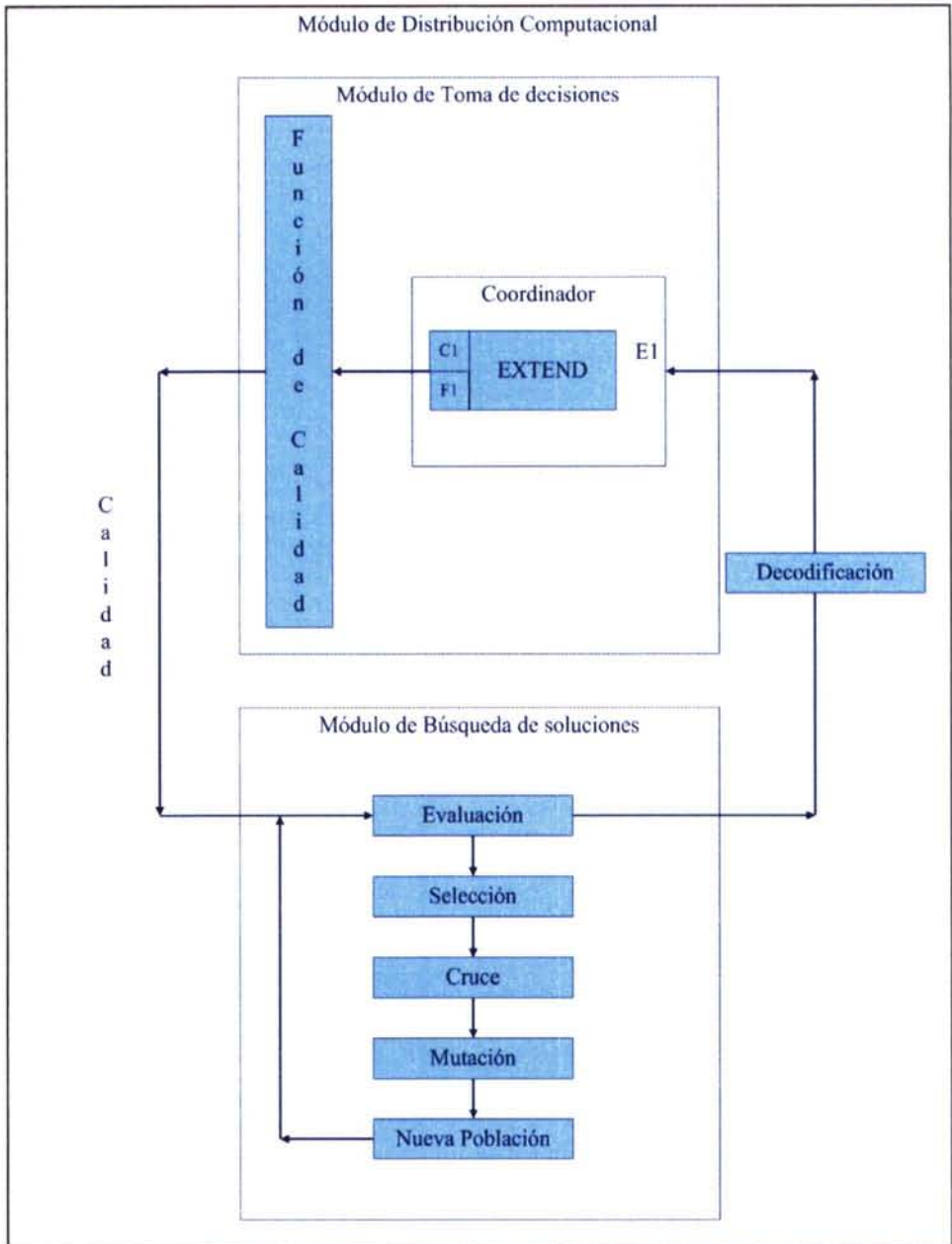


Figura 7.3: Esquema del Entorno Evolutivo de Diseño Automático implementando

El único simulador presente en el módulo de Toma de Decisiones es EXTEND, simulador comercial de sistemas discretos en donde es posible simular el comportamiento de una línea de producción industrial y, al mismo tiempo, sistemas de control electrónicos.

A continuación se detallan dos aplicaciones de la metodología evolutiva de diseño para este tipo de procesos: la optimización de morfologías y controladores para plantas industriales.

7.2.1 Optimización automática de morfologías de planta

Una de las posibilidades de este método es la optimización de la configuración morfológica de una planta de manufactura.

Una de las virtudes de este sistema es que el ingeniero es capaz de diseñar objetivos múltiples, como por ejemplo:

- Cantidad mínima de máquinas de transporte en la línea o pulmones (*buffers*)
- Capacidad mínima de dichas máquinas,
- Minimización de la energía consumida por la línea en términos de velocidades de estos elementos,
- Maximización de la cadencia de elementos de entrada,
- Cualquier combinación lineal de las anteriores con cualquier nivel de ponderación.

A continuación se expondrá un caso práctico de optimización de una línea de manufactura. En este caso una línea de montaje constituida por dos generadores de subproductos complementarios.

La distribución estadística triangular asociada a cada generador de subproducto aún siendo la misma, presenta un valor medio distinto (1 y 3). Se pretende obtener la productividad máxima de la línea de ensamblaje mediante el auxilio de bloques tipo *buffer* que evitarán, en la medida de lo posible, la aparición de cualquier bloqueo debido al asincronismo de llegadas de subproductos al módulo de ensamblaje. En la siguiente figura se muestra la disposición de los bloques que conforman la línea de ensamblaje en donde los dos bloques de color verde generan cada uno de los subproductos y el gris que presenta un triángulo azul en su interior sirve de módulo de ensamblado de las dos sublíneas que se desean diseñar.

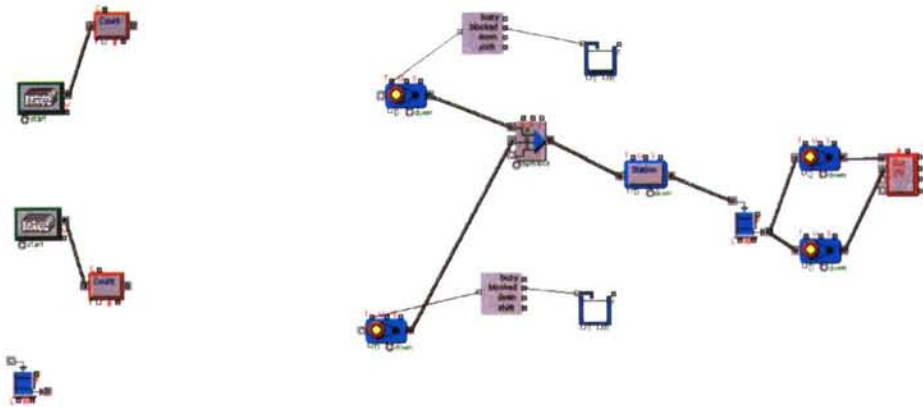


Figura 7.4: Línea de ensamblaje

Mediante evolución se pretende obtener la disposición morfológica óptima de los *buffers* de tal modo que su número y capacidad máxima sean mínimos mientras se mantiene la máxima productividad (se ensamblan todas las partes existentes del producto).

Si el número y/o capacidad de los *buffers* exceden del óptimo la planta alcanzará igualmente su máxima productividad pero el coste de la línea se incrementará innecesariamente. Por otro lado si el número y/o capacidad de los *buffers* es insuficiente aparecerán atascos en la línea y se desprenderán de la línea determinados subproductos antes del proceso de ensamblaje.

En este caso cada uno de los individuos de la población está representado por un cromosoma de longitud variable que representa dos circuitos. Inicialmente se define a cada población como una matriz dispersa cuyas filas representan el número de individuos y las columnas la capacidad de cada uno de sus *buffers*. El primer elemento cero de cada cromosoma separa los dos circuitos de cada individuo y los restantes (desde el final del segundo circuito hasta la última columna) indican el final del cromosoma. Para evitar manipular matrices dispersas se transforma esta matriz en otra sin ceros en donde se indica la posición de los valores no nulos en la matriz.

Así, si el cromosoma de un individuo es: 0,34, 5, 15, 0, 2, 5, 0,...0

El primer circuito estará representado por 3 *buffers* de capacidades 34, 5 y 15 y el segundo por 2 *buffers* de capacidades 2 y 5.

Uno de los pasos de mayor complejidad fue la definición de la función de calidad de cada uno de los individuos. Así, aunque es fácil determinar que la calidad es directamente proporcional a la productividad de la línea e inversamente proporcional al número de *buffers* y a su capacidad máxima, lo difícil es definir la ponderación de cada una de las ecuaciones que definen esta función mixta.

Una vez realizadas varias simulaciones se determinó que la función de calidad fuese:

$$sensor = \left(\frac{productividad}{500} \right) * \left(60 + \frac{30}{bloques + 1} + \frac{10}{capacidades + 1} \right)$$

$$calidad = \left(\frac{1}{sensor + 1} \right)$$

Los operadores genéticos del algoritmo genético aplicado son los básicos que se aplican para otros experimentos anteriormente mencionados con la única particularidad que manipulan cromosomas de longitud variable.

En las siguientes gráficas se muestran los distintos resultados de la productividad y las pérdidas de subproducto de cada sublínea según el caso.

En figura 7.5 se expone la apariencia del modelo en el caso de un cromosoma de baja calidad 0, 30, 29, 39, 0, 29, 46, 0,... 0

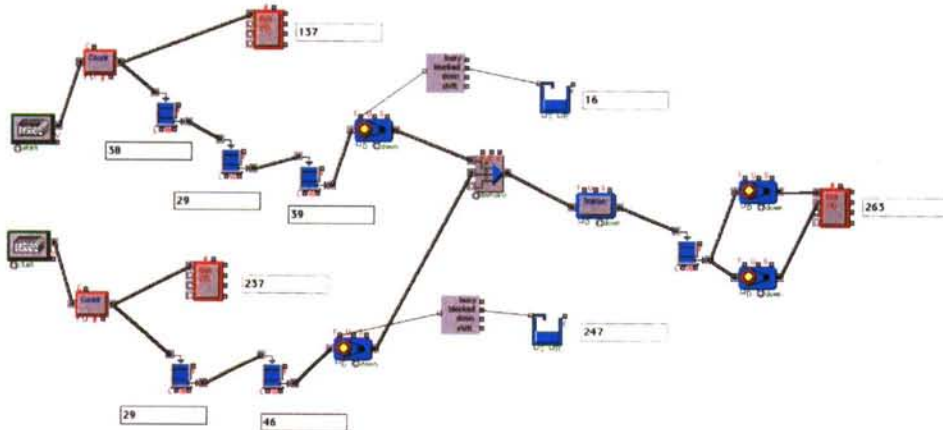


Figura 7.5: Línea de ensamble de baja calidad

Los resultados obtenidos del modelo son:

$$productividad = 263$$

$$bloques = 3 + 2 = 5$$

$$capacidades = (30 + 29 + 39) + (29 + 46) = 173$$

$$sensor = \left(\frac{263}{500} \right) * \left(60 + \frac{30}{5+1} + \frac{10}{173+1} \right) = 34,22$$

$$calidad = \left(\frac{1}{sensor + 1} \right) = 0,0284$$

Destacamos del modelo las pérdidas que se producen en la línea respecto a los 500 componentes totales: 137 para la sublínea superior y 237 para la inferior y las esperas que se producen entre componentes en el módulo de ensamble: 16 para la línea superior y 247 para la inferior.

A continuación en la figura 7.6 se muestra el modelo de mayor calidad según los parámetros de ponderación descritos

0, 7, 0, 0, 47, 45, 47, 50, 46, 47, 30, 0...0

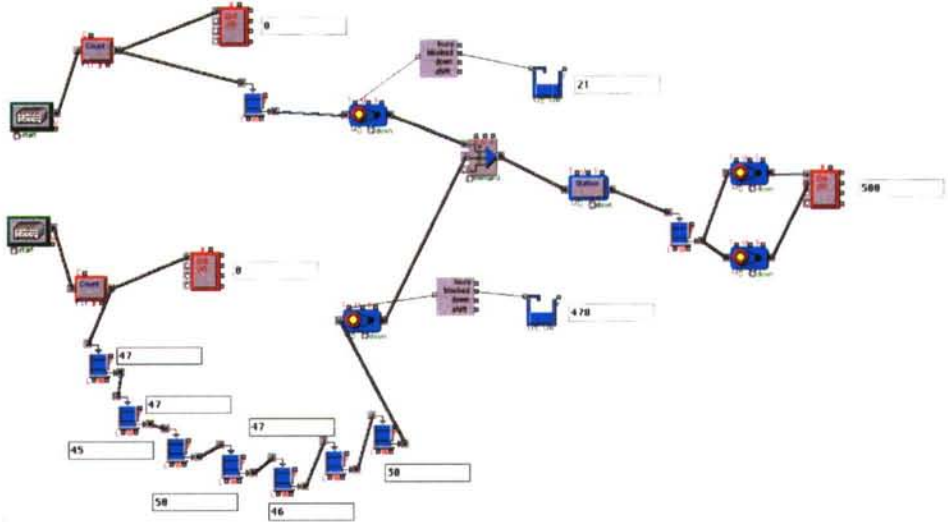


Figura 7.6: Línea de ensamblaje de mayor calidad

Los resultados obtenidos del modelo en este caso son:

$$productividad = 500$$

$$bloques = 1 + 7 = 8$$

$$capacidades = 7 + (47 + 45 + 47 + 50 + 46 + 47 + 30) = 319$$

$$sensor = \left(\frac{500}{500} \right) * \left(60 + \frac{30}{85+1} + \frac{10}{319+1} \right) = 60,38$$

$$calidad = \left(\frac{1}{sensor+1} \right) = 0,0163$$

Para la configuración correspondiente a este individuo la productividad es máxima y no existe pérdida de ninguno de los componentes. Las esperas en el módulo de ensamblaje (21 y 478) existen desde el momento en que el valor medio de las distribuciones estadísticas que modelan la cadencia de entrada de los componentes es distinto y el de la sublínea inferior (3) triplica a la superior (1). Este es el motivo por el que los *buffers* de la línea inferior superarán siempre a los de la superior en los modelos correspondientes a individuos de mayor calidad.

La conclusión básica de este apartado es que el entorno de diseño expuesto responde con buenos resultados al problema de optimización automática de morfologías de plantas desde el punto de vista de disposición y especificación de máquinas. A continuación se expondrá este mismo concepto aplicado a la síntesis de circuitos secuenciales para el control de la línea de barnizado y secado expuesta en apartados anteriores.

7.2.2 Optimización automática de controladores secuenciales

En este apartado se pretende demostrar la capacidad del entorno desarrollado para el diseño de controladores secuenciales que optimicen la productividad de una línea de barnizado y secado implantada en una industria real.

En este caso en concreto se pretende obtener el controlador que module la velocidad de los elementos de entrada y salida de un horno instalado en la planta evitando así la existencia de esperas o atascos para que, en definitiva, se obtengan valores de productividad óptima con el menor gasto energético posible.

Este macro-controlador está constituido por tres circuitos lógicos secuenciales diseñados mediante puertas lógicas nand y circuitos flip-flop tipo D. Las entradas de los circuitos corresponden a los sensores de la línea posicionados estratégicamente a lo largo de la planta y las salidas a las velocidades del rodillo de entrada y de los tapetes de entrada y salida.

Para este ejemplo se realiza en primer lugar un estudio normal aplicando herramientas software (Access, Extend) de modo independiente para la síntesis y análisis de posibles soluciones. Más adelante se hará este estudio mediante el entorno de diseño desarrollado y se presentarán sus ventajas.

7.2.2.1 Línea de Barnizado y Secado

Mediante el presente estudio se pretende analizar la funcionalidad de la línea de barnizado y secado de una empresa real. Posteriormente se aplicará la metodología de optimización diseñada en esta tesis y se realizará un estudio sobre la viabilidad de su aplicación y las ventajas e inconvenientes que este tipo de técnicas aportan a las líneas de producción. Se demostrará asimismo que las conclusiones obtenidas de la aplicación de la metodología a la optimización del control de la línea coinciden con las obtenidas a partir del estudio de la planta realizada mediante gran cantidad de simulaciones.

La línea de barnizado objeto de estudio consta de un horno vertical y dos líneas de manutención, una de entrada al horno (carga) y otra de salida del horno (descarga). El desglose pormenorizado de cada una de sus partes se muestra en la siguiente tabla.

	Longitud (m)	Velocidad (m/min)	Tiempo (s)	Unidades
Línea de Entrada				
Tapete de Inicio	4	8		
Limpiadora	1	8		
Tapete	4	8		
Robot de barnizado	5	8		

Horno de Infrarrojos	3	8		
Transporte de Entrada				
Rodillos de Entrada	1.1	8 ó 36		
Rampa de aceleración			0 a 1	
Rampa de deceleración			0 a 1	
Fotocélula				2
Tapete de entrada	4	8 ó 36		
Rampa de aceleración			0 a 1	
Rampa de deceleración			0 a 1	
Horno vertical				
Bandejas	4.5	36		92
Rotación bandejas			18	
Línea de Salida				
Tapete de Salida	5	8/10 o 36		
Rampa de aceleración			0 a 1	
Rampa de deceleración			0 a 1	
Fotocélula				2
Tapete	5	8/10		
Tapete	5	8/10		
Tapete	5	8/10		

A continuación se indica la notación utilizada para el desarrollo del estudio:

L_R Longitud de los rodillos de entrada = 1.1 m.

L_{TE} Longitud del tapete de entrada = 4 m.

L_{TS} Longitud del tapete de salida = 5 m.

L_B Longitud de la bandeja de carga del horno vertical = 4.5 m.

L_{NC} Longitud de la zona de no-carga = de 0.5 a 1.5 m.

ΔT_{AE} Rampa de aceleración en segundos a la entrada del horno = de 0 a 1 segundo.

ΔT_{DS} Rampa de deceleración en segundos a la salida del horno = de 0 a 1 segundo.

ΔL_{AE} Espacio recorrido por la carga durante la rampa de aceleración a la entrada del horno.

$$\Delta L_{AE} = \frac{8+36}{2} \cdot \frac{\Delta T_{AE}}{60}$$

ΔL_{DS} Espacio recorrido por la carga durante la rampa de deceleración a la salida del horno.

$$\Delta L_{DS} = \frac{V_S+36}{2} \cdot \frac{\Delta T_{DS}}{60}$$

V_S Velocidad lenta de la línea de salida del horno = de 8 a 10 m/min.

L_{CE} Longitud de las cargas a la entrada del horno.

L_{CS} Longitud de las cargas a la salida del horno.

ΔL_{CR} Distancia de los rodillos de entrada a la que se encuentra una carga cuando la anterior se sitúa a la entrada del horno

T_{RH} Tiempo en minutos que emplea la carga de la línea de entrada en recorrer la distancia que va desde el comienzo de los rodillos de entrada hasta la entrada del horno vertical.

T_{HE} Tiempo en minutos que el horno debe esperar, con la bandeja de carga vacía, hasta la llegada de una nueva carga por la línea de entrada.

T_{HS} Tiempo en minutos que el horno debe esperar con una carga en la bandeja de descarga hasta que el tapete de salida haya sido despejado.

En el tapete de inicio de la línea se indica mediante colores distintos el área donde el operario puede depositar los tableros para completar una carga, llamada zona de carga, y el área donde no está permitido depositar ningún tablero que se llama zona de no-carga. La zona de carga tiene una longitud igual a la máxima carga posible, es decir, igual a la longitud de una bandeja del horno vertical y que constituye una de las constantes de la instalación. La longitud de la zona de no-carga, sin embargo, constituye uno de los parámetros objeto de diseño, puesto que presenta una incidencia directa sobre la funcionalidad de la línea.

Las cargas se transportan, por tanto, enmarcadas en recintos imaginarios de dimensiones idénticas a las de una bandeja del horno y separadas una distancia constante. El operario situado en el tapete de inicio tiene la responsabilidad de enrasar el primero de los tableros que constituya una carga con la línea que delimita el comienzo de la zona de carga. De esta forma, la detección mediante fotocélulas de los comienzos de los tableros indicará inequívocamente la posición de la cabecera de la carga y de la bandeja virtual que la contiene. En la figura se ilustra la disposición de los sensores en el tramo de carga y descarga del horno.

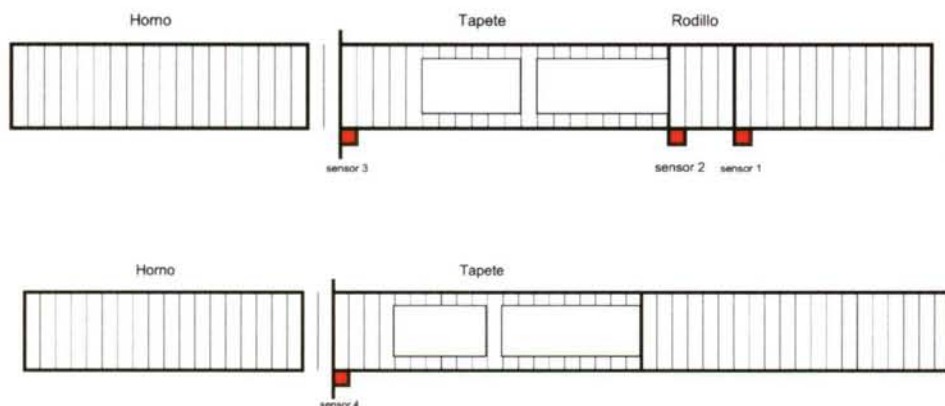


Figura 7.7: Línea de barnizado y secado

La velocidad de 8 m/min es la velocidad de operación de los distintos procesos a que son sometidos los tableros y, en particular, en el robot de barnizado constituye un

parámetro crítico puesto que de él depende el espesor de la capa de barniz o laca que se imprima.



Figura 7.8: Robot de barnizado

Dado que la carga y descarga en el horno se realizan a una velocidad de 36 m/min, es necesario incluir transportes aceleradores y deceleradores a la entrada y a la salida del horno. El transporte de entrada cumple esta función y con objeto de aumentar la cadencia de cargas de la línea, como consecuencia del acortamiento de la zona de no-carga, se ha dividido en dos partes: los rodillos de entrada y el tapete de entrada.

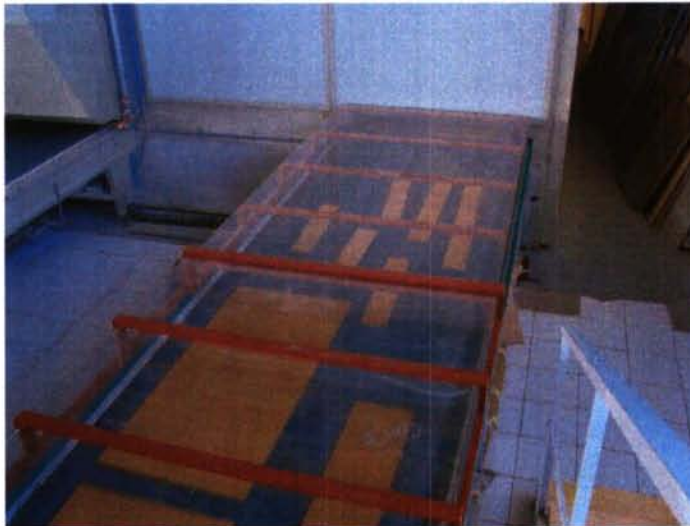


Figura 7.9: Rodillos y tapetes de entrada

En el instante en que las fotocélulas situadas al comienzo de los rodillos de entrada detectan la cabecera de una carga, el encóder del tapete da inicio al seguimiento de la posición de la carga sobre el conjunto rodillos-tapete en todo momento. Cuando la

totalidad de la carga se encuentra situada sobre este transporte, éste acelera a la velocidad de carga del horno. Un tiempo después, cuando los rodillos son liberados, éstos son decelerados a la velocidad de régimen de 8 m/min para permitir la llegada de una nueva carga. El proceso se completa cuando el tapete es liberado y retorna a su velocidad lenta.



Figura 7.10: Pulidora y robot de barnizado

Por defecto, el tapete de salida se encuentra operativo a 36 m/min, a la espera de la descarga del horno de una nueva carga. Cuando las fotocélulas situadas al final del tapete detectan la presencia de una carga, el tapete decelera a la velocidad de régimen, que aunque constante durante el funcionamiento de la línea de salida, constituye un parámetro susceptible de ser optimizado y podrá oscilar entre los 8 y los 10 m/min. Los restantes tapetes trabajarán a velocidad constante.



Figura 7.11: Salida del horno



Figura 7.12: Tapete de salida

Al final de la línea de salida, un operario recogerá las cargas. Simplificando el esquema del horno vertical, sin pérdida de generalidad, diremos que éste presenta dos estados posibles:

- Horno preparado para carga/descarga.

Este estado da comienzo cuando está ubicada una bandeja vacía en la boca de carga del horno y una bandeja ocupada en la de descarga y finaliza cuando se han completado ambas operaciones de carga y descarga.

- Rotación interna del horno.

Comprende el tiempo durante el cual el horno no se encuentra operativo para las líneas de entrada y de salida pues está realizando la rotación interna de las bandejas que consta de movimientos transversales y verticales de las mismas. El tiempo estimado de duración de este estado es de 18 segundos, según datos aportados por técnicos de la planta en estudio.

Debido a las especiales características de este estudio de optimización, marcado por las limitaciones en cuanto a los parámetros accesibles para su modificación, se concluyó ya al inicio del mismo que el parámetro de productividad no resultaba ser el mejor índice de evaluación del comportamiento de la línea. La productividad en términos de número de productos o superficie barnizada viene determinada, sobre todo, por la longitud de las cargas de entrada, su velocidad de movimiento y por la cadencia de éstas.

Consecuentemente, la productividad final será función de:

- Densidad de llenado

La densidad de llenado está relacionada con lo que los operarios llenen la zona de carga, tanto en densidad por metro lineal como en porcentaje de los 4.5 metros máximos. Este llenado, además de depender de la habilidad del operario, depende también de parámetros geométricos en la distribución de piezas que le llegan en un orden determinado y que condicionan el nivel de ocupación de la zona de carga. Optimizar esta densidad de llenado implicaría una actuación sobre la distribución de carga, bien por medio de software de optimización geométrica asociado a un puntero indicador de posición o por medio de una secuenciación previa de las piezas que llegan al operario a través, otra vez, de algún algoritmo de optimización geométrica. En cualquier caso, las limitaciones impuestas en las especificaciones del presente estudio dejan estas consideraciones fuera del mismo.

- Velocidad de movimiento de las cargas

Es posible plantear un ajuste de la velocidad de los transportes de entrada y salida al horno para mejorar la productividad de la línea. Esta solución permite incluso aumentar la cadencia de tableros de entrada al horno e incrementar así el rendimiento del sistema.

- Cadencia de cargas

Mejorar la cadencia de las cargas, es decir, cuantas cargas circulan por unidad de tiempo a una velocidad de línea determinada, implica reducir la distancia entre

zonas de carga. Obviamente, si uno no considerase ningún otro elemento, la cadencia óptima resultaría cuando la distancia entre las zonas de carga es cero. El problema que surge es que se producirían múltiples colisiones que implicarían paradas de la línea y pérdidas de material, lo que incidiría muy negativamente en la productividad.

Como consecuencia de esto, se ha determinado que los elementos relevantes a estudiar para determinar los parámetros óptimos del sistema son los retrasos y paradas de la línea. Esto es, el objetivo se ha convertido en obtener condiciones para poder permitir la mayor cadencia de cargas y velocidad de la línea con la condición de que no existan paradas por acumulación. La determinación de las causas que originan estos retrasos y paradas se convierte entonces en el eje alrededor del cual gira todo el problema pues constituye el único inconveniente grave que puede derivarse de la reducción de la zona de no-carga y el aumento de la velocidad de los elementos de transporte.

Un primer enfoque del problema consiste en poner los medios para que las paradas sean absorbidas por el sistema y no lleguen a afectar a elementos críticos como es el robot de barnizado. Estas soluciones pueden ser:

1. Inclusión de un pulmón o *buffer* entre el horno de infrarrojos y el transporte de entrada que permita almacenar temporalmente aquellas cargas que no pueden ser atendidas por el horno vertical, es decir, no pueden ser cargadas en éste. La capacidad de este pulmón no tendría que ser de más de 1 ó 2 unidades puesto que dada la velocidad de la línea el operario dispone de suficiente tiempo para reaccionar y permitir la descarga del pulmón.
2. Alargar el transporte de entrada acelerador en su conjunto para provocar un mayor distanciamiento entre cargas consecutivas y una mayor duración de la operación de aceleración y carga en el horno. Este tiempo mayor no presenta incidencia alguna en la productividad y si en la ausencia de paradas del horno y de la línea, tal como se verá más adelante.

Dado que el presente estudio consiste en la optimización del funcionamiento de una línea existente sin intervenir en su diseño morfológico, las dos soluciones anteriores no pueden ser consideradas. Nos resta, entonces, profundizar en las causas que conducen a paradas y retrasos en la línea.

Desarrollaremos a continuación dos planteamientos distintos. El primero estudia las paradas de la línea como consecuencia, exclusivamente, de las longitudes de las zonas de no-carga y de las longitudes de las cargas que llegan al horno. A este primer enfoque lo denominaremos el Problema de la Carga del Horno. El segundo aborda el problema como consecuencia de retrasos o retardos en la operación de descarga del horno. Lo denominaremos el Problema de la Descarga del Horno.

- El Problema de la Carga del Horno

Supongamos que dos cargas viajan por la línea de entrada. La primera de ellas de longitud $(L_{CE})_{\max}$ y la segunda de longitud $(L_{CE})_{\min}$, tal que $(L_{CE})_{\max} > (L_{CE})_{\min}$. Cuando la primera de ellas se encuentre en la entrada del horno, la segunda de las cargas se encontrará a una distancia tal que el tiempo que emplee en llegar a la entrada del horno debe ser igual o mayor que el tiempo de carga del horno más el tiempo de rotación interna del horno, suponiendo que no existe retraso alguno del horno debido a descargas

no completas o no realizadas. Centr ndonos en el punto cr tico donde ambos tiempos son iguales, obtenemos:

$$\frac{L_B + L_{NC}}{8} - \frac{(L_{CE})_{max}}{8} - \frac{L_R + L_{TE} - (L_{CE})_{max} - \Delta L_{AE}}{36} - \frac{\Delta T_{AE}}{60} + K$$
$$K + \frac{(L_{CE})_{min}}{8} + \frac{L_R + L_{TE} - (L_{CE})_{min} - \Delta L_{AE}}{36} + \frac{\Delta T_{AE}}{60} = \frac{L_B}{36} + \frac{18}{60}$$

de donde:

$$(L_{CE})_{min} = (L_{CE})_{max} - L_B - \frac{4.5 \cdot L_{NC}}{3.5} + \frac{36 \cdot 18}{3.5 \cdot 60}$$

La interpretaci n de este resultado es la siguiente:

Dada una longitud de la zona de no-carga determinada, para cada carga que entre en el horno existir  una longitud m nima de carga $(L_{CE})_{min}$ para la carga inmediatamente posterior que garantiza que la l nea no va a sufrir una parada si la longitud de dicha carga es igual o superior.

En las figuras que aparecen a continuaci n se ven los valores que toma $(L_{CE})_{min}$ para distintos valores de $(L_{CE})_{max}$ y para distintas longitudes de no-carga.

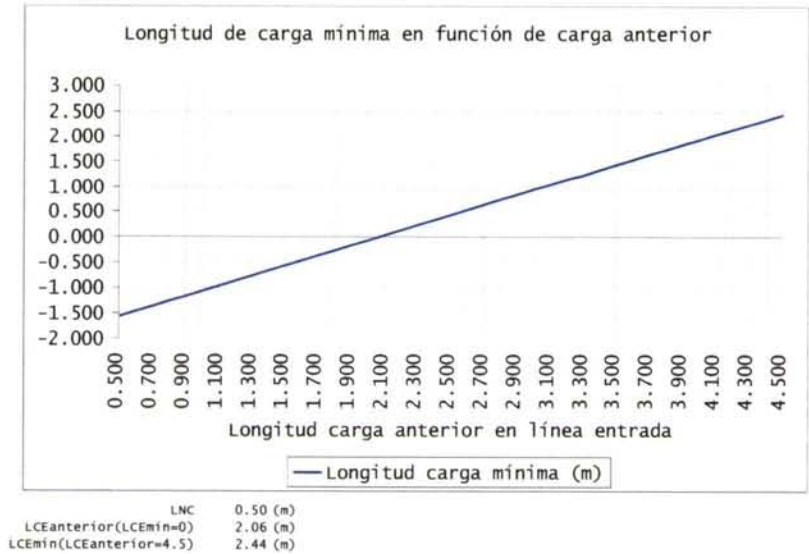


Figura 7.13: Longitud de carga m nima

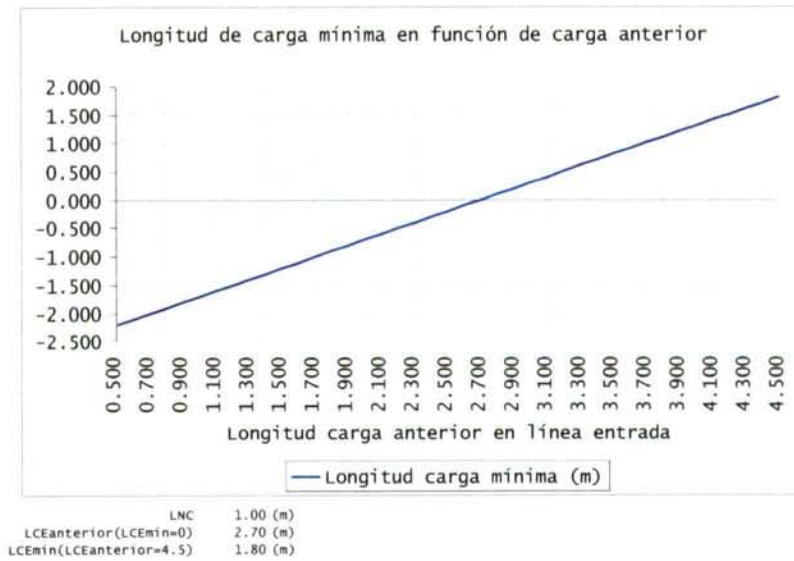


Figura 7.14: Longitud de carga mínima

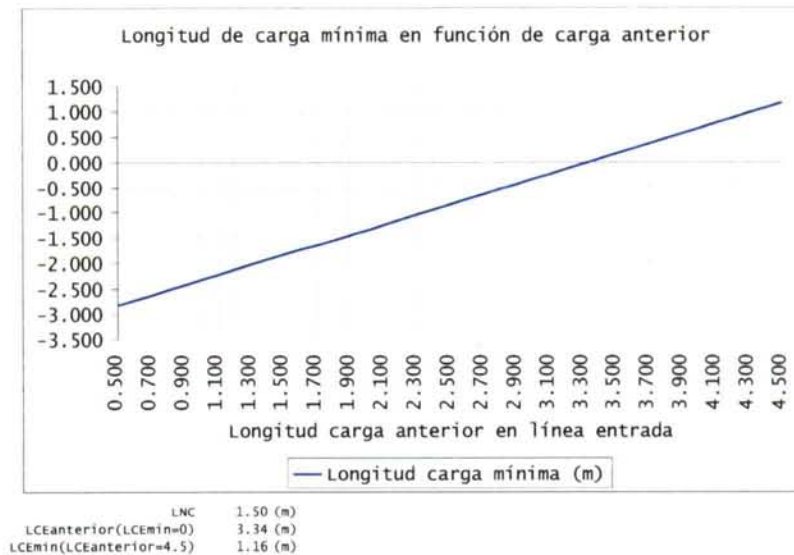


Figura 7.15: Longitud de carga mínima

Especial interés presenta el cálculo de $(L_{CE})_{min}$ cuando $(L_{CE})_{max} = L_B = 4.5$ m. La siguiente figura representa un resumen de todo el abanico de gráficas que, como las anteriores, se pueden calcular para longitudes de no-carga que van desde lo 0.5 m hasta los 1.5 m. La zona sombreada indica al margen de longitudes de cargas, para una zona de no-carga dada, dentro del cual en ningún caso se producirá una parada debida a problemas en la carga del horno, siempre que dichas cargas se encuentren en su totalidad dentro de esta zona. Para cargas fuera de esta zona que estén ubicadas por encima o sobre la recta de $(L_{CE})_{min}$ tampoco se producirán paradas debidas a problemas en la carga del horno. En los restantes casos es necesario estudiar cada carga en función de la inmediatamente anterior para concluir si se va a producir una parada o no.

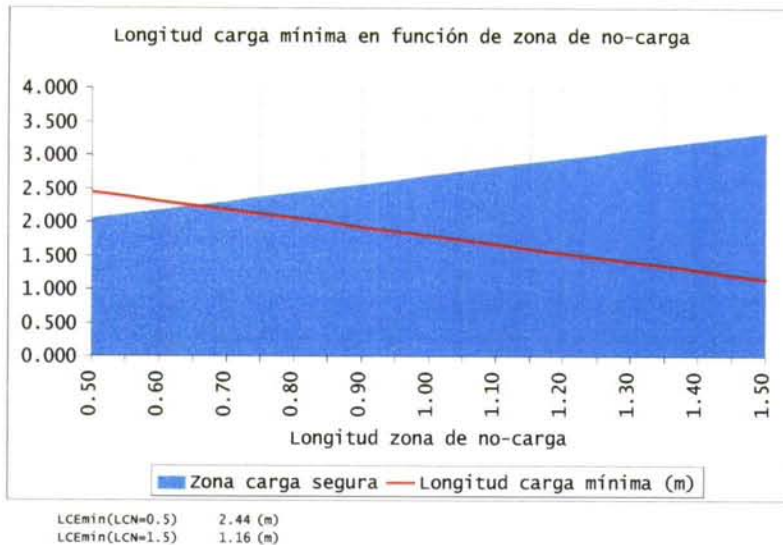


Figura 7.16: Longitud de carga mínima

- El Problema de la Descarga del Horno

Cuando el horno vertical se encuentra preparado para realizar las operaciones de carga/descarga pero el tapete de salida no ha sido evacuado en su totalidad por la carga descargada momentos antes, el horno retrasa la operación de descarga un tiempo igual al empleado en despejar dicho tapete. A estas esperas las denominaremos, a partir de ahora, Retrasos en la Descarga del Horno.

Supongamos, ahora, que ya ha sido cargada completamente una bandeja en el horno vertical y que la descarga no ha finalizado todavía. Esto puede ser debido a que la descarga no ha finalizado o que el tapete de salida no está libre y la descarga no se ha iniciado. En este caso, el horno no podrá evolucionar al estado de rotación interna de las bandejas mientras la descarga no haya concluido. Este tiempo lo denominaremos Retraso en la Rotación del Horno.

Como puede apreciarse, estos retrasos así definidos no coincidirán necesariamente en el tiempo. En realidad, el número de retrasos en la descarga es superior al número de retrasos en la rotación.

Además, no podemos concluir que la aparición de estos retrasos ocasione una parada en la línea. Ni siquiera podemos afirmar que un número elevado de ellos provoque una parada de la línea pero el estudio analítico que desarrollamos a continuación y los resultados de las simulaciones realizadas confirman que cuando una parada de la línea no tiene su origen en un problema en la carga del horno, siempre están presentes un elevado número de retrasos en la rotación del horno y de retrasos en la descarga del horno.

- Retrasos del Horno vs. Zona de No-Carga y Velocidad Línea de Salida

Supongamos un conjunto indefinido de cargas de longitud L_{CE} que son transportadas por la línea de entrada y un conjunto indefinido de cargas de longitud L_{CS} que son descargadas de forma continua del horno vertical.

Cuando una carga se sitúa a la entrada del horno, la inmediatamente anterior no habrá llegado todavía al comienzo de los rodillos de entrada y se encontrará a una distancia de éstos igual a:

$$\Delta L_{CR} = L_B + L_{NC} - 8 \cdot T_{RH}$$

$$T_{RH} = \frac{L_{CE}}{8} + \frac{L_R + L_{TE} - L_{CE} - \Delta L_{AE}}{36} + \frac{\Delta T_{AE}}{60}$$

Entonces, si no se produce ningún tipo de retardo en el funcionamiento del horno, el tiempo que emplee esta carga en situarse a la entrada del horno debe ser igual al empleado en introducir completamente la carga anterior más los 18 segundos que tarda el horno en rotar las bandejas internamente. Esto nos lleva a la siguiente igualdad:

$$\frac{\Delta L_{CR}}{8} + T_{RH} = \frac{L_B}{36} + \frac{18}{60}$$

El que todas las cargas de entrada se hayan considerado iguales hace que, en la anterior igualdad, T_{RH} desaparezca como variable y, por tanto, también desaparezca como variable la longitud de las cargas de entrada. Podemos despejar, entonces, la longitud de la zona de no-carga que hace que el sincronismo antes descrito sea posible y obtendremos:

$$\frac{L_B + L_{NC}}{8} = \frac{L_B}{36} + \frac{18}{60}$$

$$L_{NC} = 8 \cdot \left(\frac{L_B}{36} - \frac{4.5 \cdot L_B}{36} + \frac{18}{60} \right)$$

$$L_{NC} = 2.4 - \frac{3.5 \cdot L_B}{4.5}$$

Es inmediato observar que, para la longitud de la bandeja de 4.5 m, la anterior ecuación arroja un resultado negativo para la longitud de la zona de no-carga, $L_{NC} = -1.1$ m. Esto se traduce en que, para valores de la zona de no carga que van entre los 0.5 m y los 1.5 m, siempre se cumplirá que:

$$\frac{L_B + L_{NC}}{8} > \frac{L_B}{36} + \frac{18}{60}$$

$$\frac{L_B + L_{NC}}{8} = \frac{L_B}{36} + \frac{18}{60} + \Delta T_{HE}$$

es decir, el horno siempre esperará un tiempo ΔT_{HE} con la bandeja de carga vacía a que llegue una nueva carga, siempre bajo el supuesto de que no haya ocurrido ningún tipo de retención durante la descarga del mismo horno.

Para que estas retenciones no tengan lugar las cargas de la salida, que se han considerado todas iguales, deben abandonar el tapete de salida en los 18 segundos de rotación interna del horno, puesto que las operaciones de entrada y salida de horno consumen el mismo tiempo. Esto nos lleva a la siguiente igualdad:

$$\frac{18}{60} = \frac{L_{TS} - L_{CS} - \Delta L_{DS}}{36} + \frac{L_{CS}}{V_S} + \frac{\Delta T_{DS}}{60}$$

de donde:

$$L_{CS} = \frac{36 \cdot V_S}{36 - V_S} \cdot \left(\frac{18 - \Delta T_{DS}}{60} - \frac{L_{TS} - \Delta L_{DS}}{36} \right)$$

Cualquier carga de mayor longitud que la calculada anteriormente provocará retenciones de las cargas en la bandeja de salida del horno en espera de que el tapete de salida quede completamente despejado y pueda acelerarse a la velocidad de 36 m/min.

Para una rampa de deceleración del tapete de salida de 0.5 segundos y una velocidad lenta de la línea de salida de 10 m/min, obtendremos una $L_{CS} = 2.19$ m.

Es decir, para cargas iguales a la salida del horno de 2.19 m o menores el funcionamiento del horno será el óptimo, es decir, sin retenciones y no se producirán paradas ni colisiones en la línea de entrada para un conjunto de cargas de igual longitud. Este resultado es independiente de la longitud de las cargas en la línea de entrada y de la longitud de la zona de no-carga entre las bandejas.

Como es absurdo el limitar la longitud máxima de las cargas a la salida, fijaremos ésta a la máxima posible, es decir, la longitud de la bandeja $L_B = 4.5$ m. Entonces, una vez evacuada la carga de la bandeja de descarga del horno, aquélla despejará el tapete de salida en el siguiente tiempo:

$$T_D = \frac{L_{TS} - L_B - \Delta L_{DS}}{36} + \frac{L_B}{V_S} + \frac{\Delta T_{DS}}{60}$$

Además, por lo ya dicho, tenemos la certeza de que:

$$T_D > \frac{18}{60}$$

$$T_D = \frac{18}{60} + \Delta T_{HS}$$

o lo que es lo mismo, de encontrarse una nueva carga situada sobre la bandeja de salida del horno, ésta tendrá que retenerse $\Delta T_{HS} = 0.167$ minutos, dada la velocidad de salida y la rampa de deceleración indicadas anteriormente.

Como ya habíamos apuntado, la geometría de las bandejas y de la línea de entrada obliga a retardos en la carga del horno de ΔT_{HE} minutos así que, se conseguirá el mayor aprovechamiento del horno, que se traduce en una ausencia de atascos en la línea de entrada, cuando se cumpla:

$$\Delta T_{HE} = \Delta T_{HS}$$

$$\frac{L_B + L_{NC}}{8} = \frac{L_B}{36} + \frac{18}{60} + \Delta T_{HS}$$

$$L_{NC} = 8 \cdot \left(\frac{18}{60} + \Delta T_{HS} - \frac{3.5 \cdot L_B}{36} \right)$$

De aquí obtendremos $L_{NC} = 0.24$ m, que debemos interpretar de la siguiente manera: la descarga continuada de cargas de longitud máxima, es decir, 4.5 m producirán retenciones en el funcionamiento normal del horno que provocarán una colisión de las cargas en la entrada del horno, supuestas todas iguales, siempre que la distancia de no-carga sea inferior a 0.24 m (para una rampa de deceleración de 0.5 segundos y una velocidad de salida de 10 m/min). En caso de ser mayor o igual, nunca se producirá colisión alguna. Este resultado es independiente de la longitud de las cargas de entrada. La prontitud con que se observe una colisión en la línea de entrada será tanto mayor cuanto menor sea la longitud de la zona de no-carga.

En las figuras que se presentan a continuación se observan los límites mínimos de las longitudes de las zonas de no-carga en función de la velocidad de la línea de salida y la longitud de las cargas en la línea de salida. Obviamente, para una velocidad de la línea de salida dada, debemos prestar la máxima atención al caso más desfavorable, es decir, aquél en el que las cargas en la línea de salida son de longitud máxima igual a 4.5 m.

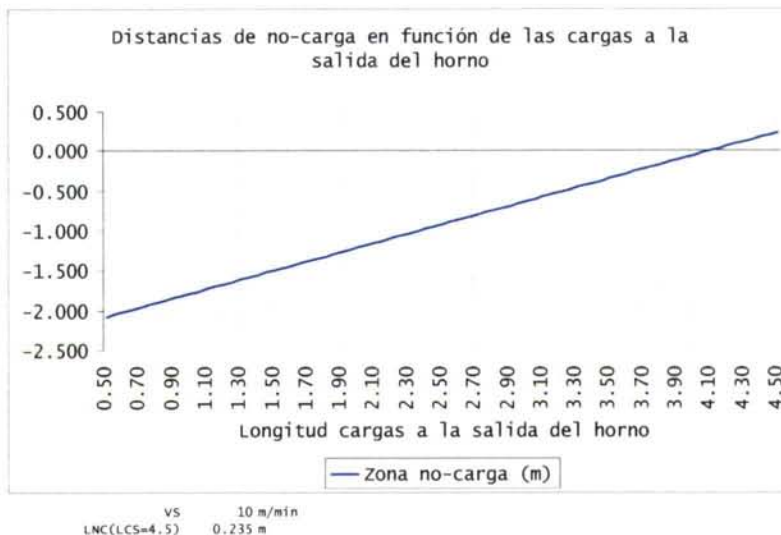


Figura 7.17: Distancias de no-carga

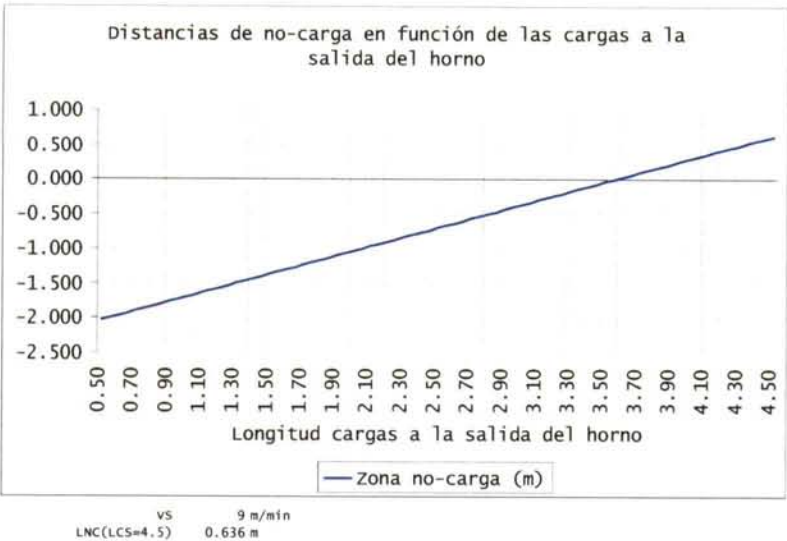


Figura 7.18: Distancias de no-carga

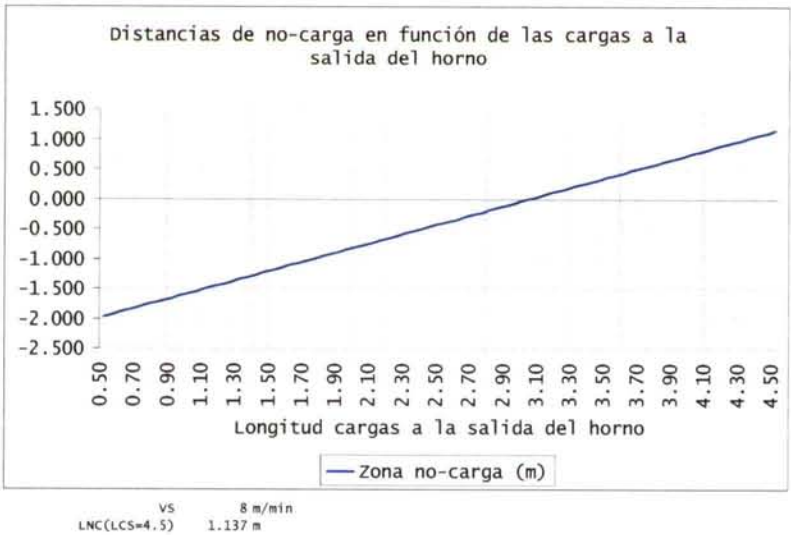


Figura 7.19: Distancias de no-carga

Del estudio de estas gráficas podemos observar que si pretendemos disminuir tanto como sea posible la longitud de la zona de no-carga no nos podemos limitar a una velocidad de la línea de salida de 8 m/min y que el incremento de 8 a 10 m/min de esta velocidad supone una disminución de la zona de no-carga de 0.9 m.

El análisis cuantitativo que acabamos de desarrollar responde a unos supuestos muy restrictivos en lo que respecta a las longitudes de las cargas de entrada y de salida del

horno al considerar que éstas sean todas iguales. Sin embargo de los resultados obtenidos se extraen las siguientes conclusiones:

- La velocidad del tapete de salida es crucial para el estudio de optimización de la línea.
- El sistema es lo suficientemente complejo como para que su modelado y simulación aporten nuevos puntos de vista para la resolución del problema en cuestión.
- Es necesario modelar estadísticamente la generación de tableros en la línea evitando hacer hipótesis demasiado restrictivas lejos del comportamiento real del proceso.

En el modelo que se comentará a continuación se reproducirán las hipótesis de partida del estudio y se observará el rendimiento de la planta en función de la velocidad de salida de la línea y de la longitud de la zona de no-carga.

7.2.2.2 Modelado

La realización de pruebas de optimización experimentando con la planta real supondría grandes pérdidas económicas si fuese necesario parar la producción. También es deseable evitar la posibilidad de averías en el horno vertical o en cualquiera de los equipos involucrados en el sistema. Por este motivo y por el hecho que un estudio matemático completo presentaría una excesiva complejidad se optó por el modelado.

Al tratarse de un sistema con un funcionamiento relativamente complejo, se seleccionó una herramienta de simulación específica para el diseño de plantas industriales: EXTEND. Esta herramienta es un entorno orientado a objetos para el diseño, análisis y documentación de procesos discretos. Sin embargo, aún cuando se especializa en este tipo de simulaciones, fue necesario crear nuevos módulos mediante programación para emular el comportamiento del sistema.

Por tratarse ésta de una línea ya diseñada con anterioridad, el estudio se centró en la optimización del funcionamiento de la misma, en términos de maximización de la producción y minimización de costes, sin pretender modificar la disposición de la misma. El único objetivo fue la modificación de ciertos parámetros característicos de los elementos que configuran la línea para proporcionar su mayor rendimiento económico. Como punto de partida de esta optimización es necesario parametrizar el sistema en función de sus características funcionales y morfológicas.

Los puntos clave de esta etapa son el modelado del horno vertical y de la problemática intrínseca relacionada con la entrada y salida de tableros. También requirió de un esfuerzo adicional modelar los diversos tipos de atascos y los deslizamientos en el tándem de entrada rodillo-tapete.

A continuación se muestra el modelo de la línea de barnizado de la planta en estudio fragmentada en varias gráficas para facilitar su visualización.

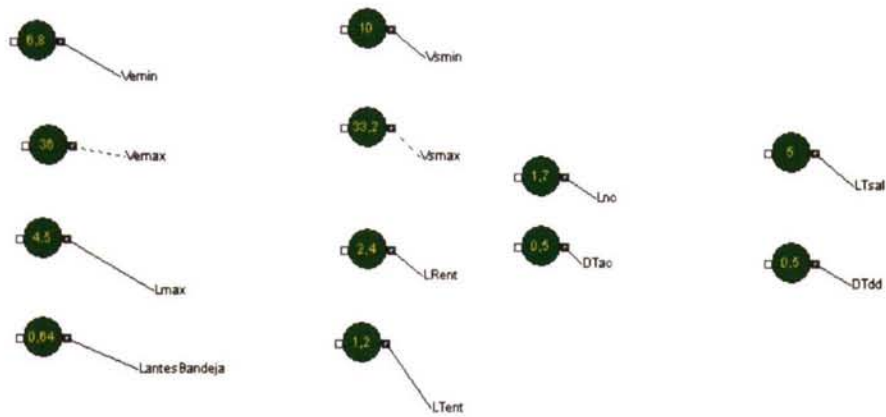


Figura 7.20: Asignación de parámetros constantes.

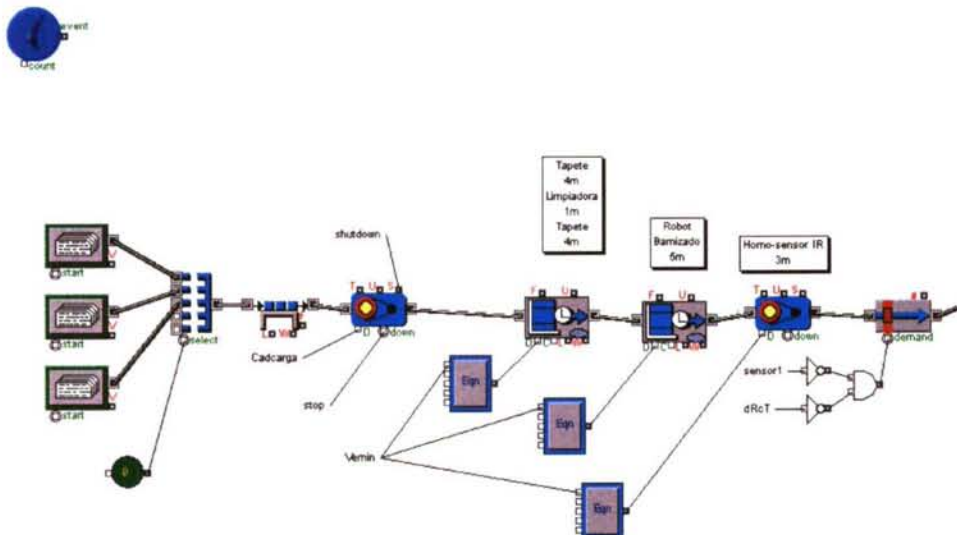


Figura 7.21: Línea de entrada. Robot de barnizado.



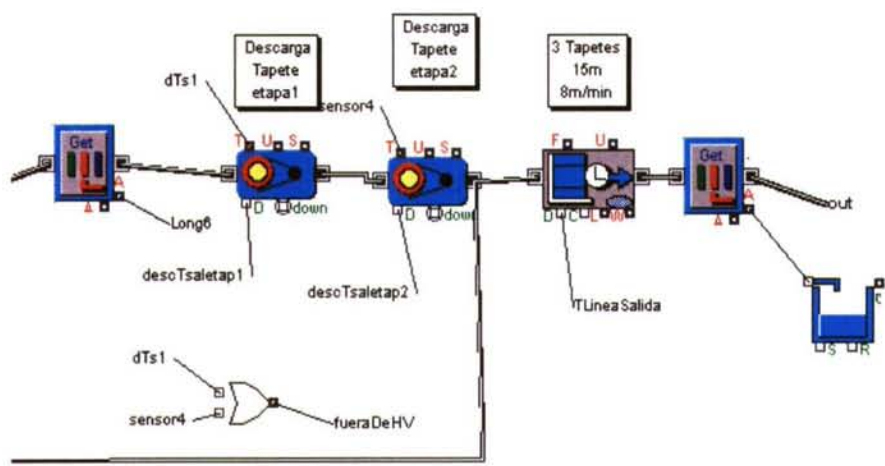


Figura 7.25: Descarga del tapete de salida.

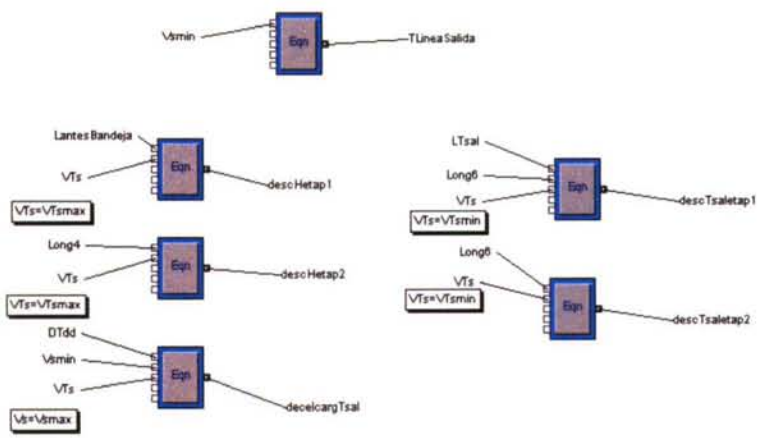


Figura 7.26: Asignación de parámetros variables.

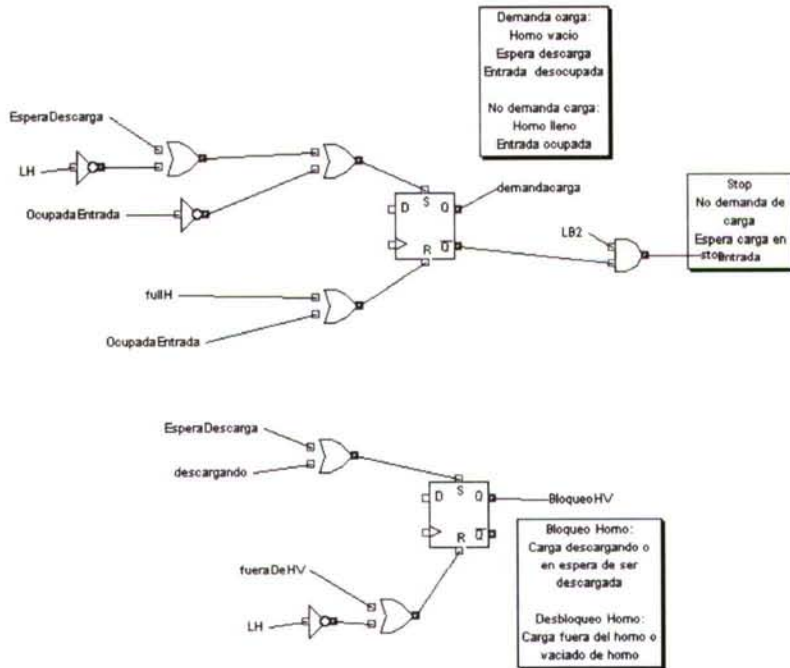


Figura 7.27: Control de la demanda de carga y bloqueo del horno

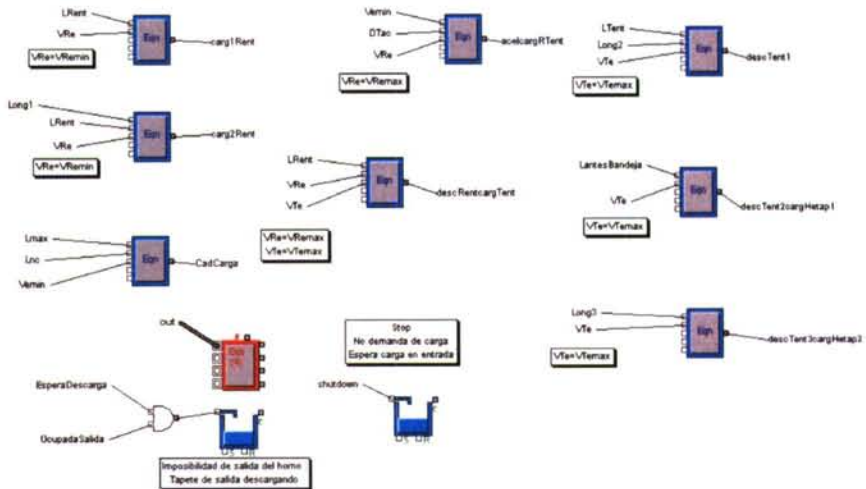


Figura 7.28: Cálculo de parámetros variables. Atascos.

La solución óptima sería aquella que minimizando la zona de no carga, proporcionase mayor productividad y menor número de atascos con la menor energía posible. Para ello se convino en realizar distintas simulaciones en función de los parámetros considerados como críticos.

7.2.2.3 Resultados del estudio tradicional

Detallamos a continuación los resultados de 252 simulaciones ejecutadas, las más relevantes de un conjunto total de más de 2.000.

Estas simulaciones tienen como objetivo ilustrar las paradas y los retrasos de la línea como consecuencia del problema en la descarga del horno, así como los retrasos producidos en su rotación. Para ello se modeló el generador de cargas como una distribución triangular de valores 4.5 (máximo) y 0.5 (mínimo) que reproducen la cadencia de aportación de tableros a la línea.

Los datos básicos del conjunto de simulaciones que se exponen se describen en la siguiente tabla:

Tiempo de simulación	2400 minutos
Generador de cargas	Aleatorio
Modelo generador de cargas	Distribución triangular
Longitud mínima de carga	de 0.4 a 4.5 m
Zona de no-carga	0.5/1/1.5 m
Velocidad de la línea de salida	8/10 m/min
Rampa aceleración transporte entrada	0.5 segundos
Rampa deceleración transporte entrada	0.5 segundos
Rampa aceleración tapete salida	0.5 segundos
Rampa deceleración tapete salida	0.5 segundos
Número total de simulaciones	252 simulaciones

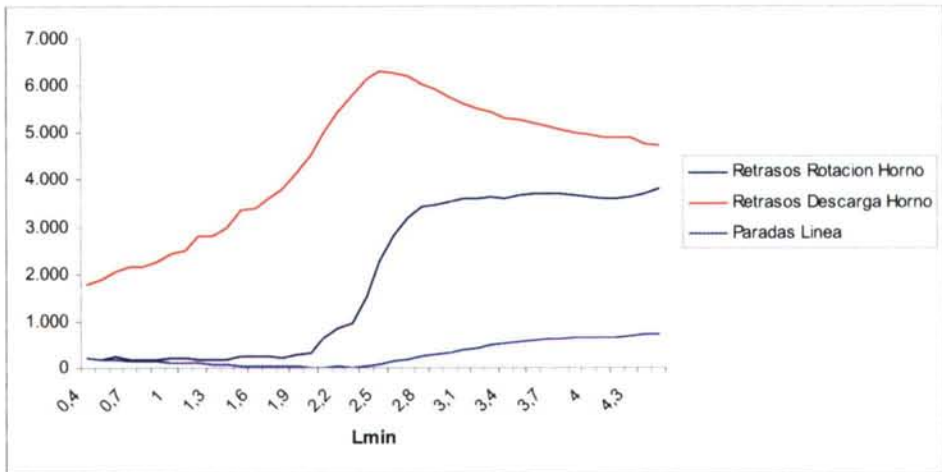
- Caso 1

$V = 8 \text{ m/min}$

rampa decel=0.5

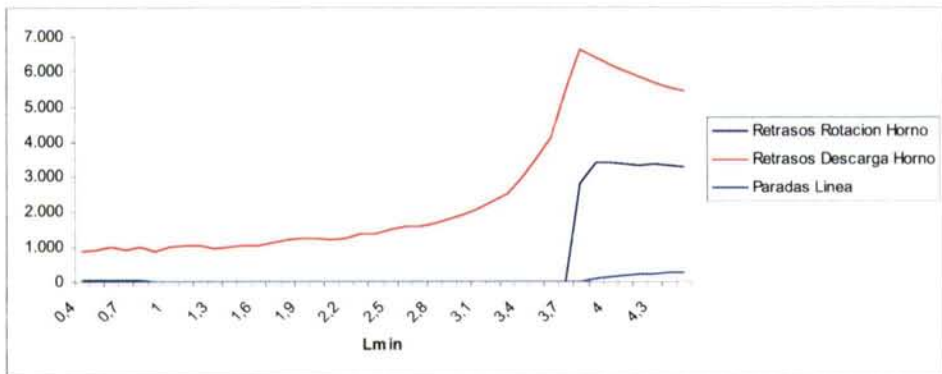
Duración = 2400 min

Numero simulaciones=40



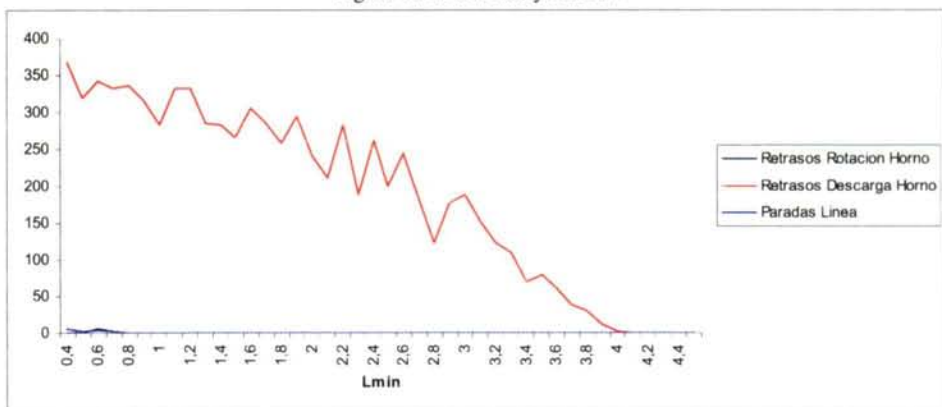
$L_{nc} = 0.5 \text{ m}$

Figura 7.29: Retrasos y Paradas



$L_{nc} = 1 \text{ m}$

Figura 7.30: Retrasos y Paradas



$L_{nc} = 1.5 \text{ m}$

Figura 7.31: Retrasos y Paradas

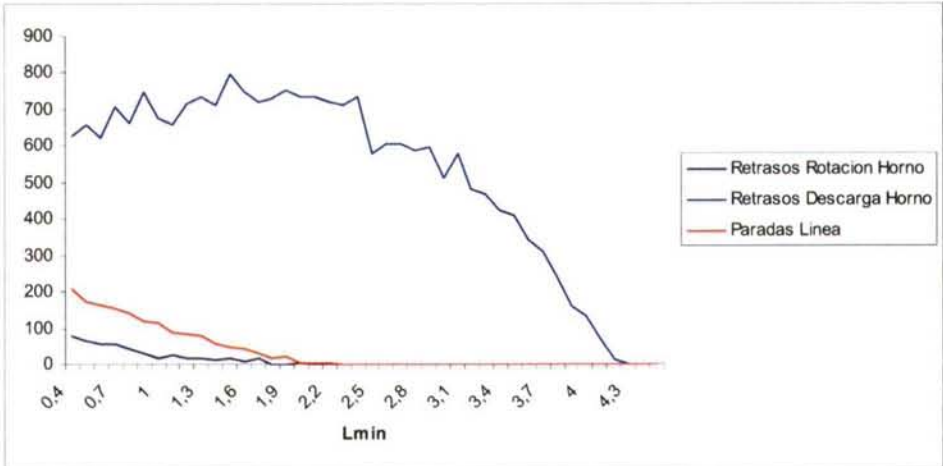
• Caso 2

$V = 10 \text{ m/min}$

rampa decel=0.5

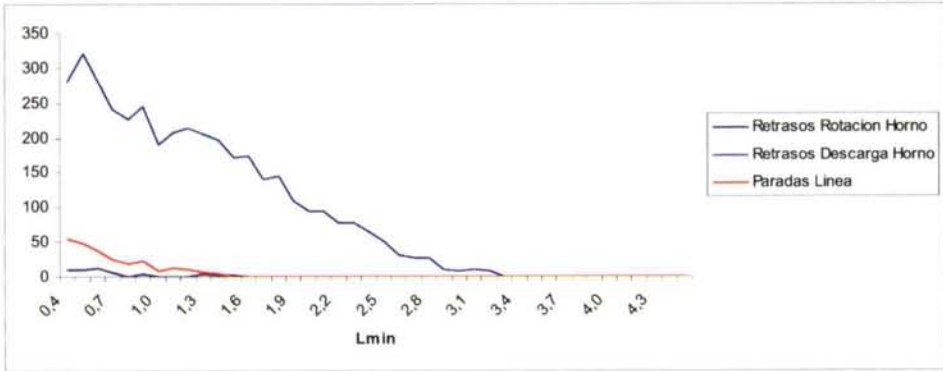
Duración = 2400 min

Numero simulaciones=40



$Lnc = 0.5 \text{ m}$

Figura 7.32: Retrasos y Paradas



$Lnc = 1 \text{ m}$

Figura 7.33: Retrasos y Paradas

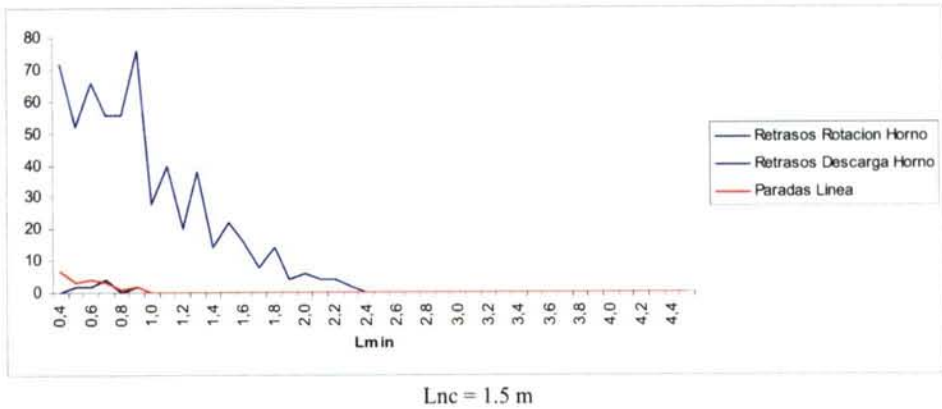


Figura 7.34: Retrasos y Paradas

A la vista de las tablas y gráficas se llegó a la conclusión de que la mejor solución sería aumentar la velocidad máxima del tapete de salida. De este modo con un cambio mínimo y sin realizar modificaciones morfológicas en la planta se evitan paradas y retrasos en la misma y el consecuente incremento de su productividad.

Esta productividad puede incrementarse aun más si se acorta la zona de no carga. En este caso y según se puede observar de la tabla cuando la velocidad del tapete de salida se ajusta a 10m/min y la zona de no carga es de 1.5 m se asegura la inexistencia de atascos en la línea para tableros de más de 1 m. Si la longitud de esta zona es de 1m los tableros de más de 1.6m no presentan problemas en la línea y para 0.5m a partir de los 2.4m.

Además de esto es de destacar que como se había indicado en el análisis matemático anterior para una velocidad del tapete de salida de 10m/min y cargas de longitud 4.5 m, no se produce ningún atasco para una zona de no-carga de 0.5, 1 o 1.5 m.

Independientemente de lo anterior, existen medidas software y hardware que se pueden implementar para prevenir la ocurrencia de estas paradas consideradas como graves.

Utilizando el encoder de la bandeja de entrada en conjunción con una fotocélula a la salida del horno el autómatas de control de línea podría anticipar las situaciones mencionadas y, antes de que ocurriese la colisión, tomar medidas tales como:

- Realizar modificaciones en línea de los parámetros y tiempos de aceleración de la secuencia de bandejas de forma que se recupere la sincronización.
- Realizar microparadas en la línea para recuperar la sincronización. Estas microparadas serían tiempos que no llevarían a problemas en la máquina de barnizado, pero que, a base de secuenciarlos adecuadamente permitirían recuperar el sincronismo.

7.2.2.4 Conclusiones del diseño tradicional

En el modelado inicial y mediante técnicas no evolutivas se llegó a la conclusión de que la optimización del sistema pasaba por el aumento de la velocidad máxima del tapete de

salida (10m/s). La simplicidad de adoptar esta solución sin la incorporación de nuevos elementos reforzó aun más lo conveniente de esta medida.

A partir de aquí se nos plantean dos posibles aplicaciones de diseño automático en plantas industriales. Por un lado la optimización paramétrica de la línea (anteriormente se habían realizado simples barridos) y, por otro, el diseño y optimización de controladores. La consecución de estas dos aplicaciones da lugar a una metodología donde una vez realizado el modelo paramétrico de una planta inicial básica, es capaz de generar el circuito lógico secuencial mínimo que optimice su productividad.

En relación a esto último se pretendió crear un sistema de control en planta de tal modo que en función de la sensorización establecida en el modelo, se actuase sobre los motores de los rodillos y tapetes evitando posibles atascos (también derrapes) y buscando la máxima productividad con el menor gasto energético posible.

Para sintetizar el controlador se echó mano de una codificación binaria en árbol. Con este método se optimizó el circuito lógico que con el menor número de puertas fuese capaz de aligerar la producción lo máximo posible.

Para la síntesis de controladores secuenciales de plantas industriales se utilizaron técnicas evolutivas aplicadas a sistemas electrónicos digitales constituidos por puertas lógicas y memorias. El objetivo que se persiguió se basaba en la interconexión de estos elementos básicos para la síntesis de un circuito lógico secuencial que controlase el funcionamiento de determinados actuadores de la planta en función de las señales procesadas por los sensores distribuidos en la misma. En el siguiente apartado se hará un estudio del estado del arte de este tipo de técnicas que forman parte del Hardware Evolutivo.

7.2.2.5 Estado del Arte en Hardware Evolutivo

Tradicionalmente el diseño de sistemas electrónicos se ha basado en gran medida en la experiencia y la aplicación de diversas técnicas, directrices y principios de diseño. De entre las herramientas de diseño digital con más repercusión se encuentran los Mapas de Karnaugh [Karnaugh, 53] y el método de Quine-McCluskey [Quine, 55], [McCluskey, 56]. En ambos casos, aunque su interfaz meramente gráfica permite cierta libertad, presentan importantes limitaciones cuando intentan dar solución a circuitos de alta complejidad. Otra limitación exclusiva en este caso al método Quine-McCluskey es la incapacidad de modelar componentes de mayor complejidad como por ejemplo la compuerta XOR. Espresso y MIS-II son otros algoritmos de gran eficiencia que han sido modelados y automatizados por ordenador y que permiten mediante una serie de heurísticas tratar con diseños electrónicos digitales relativamente complicados [Brayton, 84, 87].

El concepto que recibió el nombre de hardware evolutivo consiste en el diseño y simulación evolutivos de circuitos mediante la aplicación de algoritmos genéticos y que permite una mayor exploración del espacio de búsqueda. George J. Friedman fue pionero en la aplicación de técnicas evolutivas a la robótica y propuso un mecanismo automático para el diseño y simulación de circuitos mediante su codificación cromosómica y la aplicación de operadores de selección, cruce y mutación [Friedman, 56, 59].

En función del proceso de evaluación, se definen tres implementaciones distintas [Kalganova, 00a]:

- Extrínseca: todos y cada uno de los circuitos electrónicos candidatos son sintetizados según un modelo de simulación.
- Intrínseca: la solución al circuito electrónico a diseñar se implementa y evalúa mediante dispositivos reconfigurables.
- Mixtrínseca: las poblaciones de soluciones parciales son mixtas, es decir, compuestas de simulaciones y dispositivos reconfigurables. Este tipo de implementación es la menos habitual [Zebulum, 00].

En función del proceso de evolución existen otros enfoques como son el diseño a nivel de puertas lógicas básicas AND, OR y NOT [Higuchi, 96] y el diseño a nivel de funciones que pretende construir a partir de puertas lógicas básicas nuevos módulos como multiplexores, sumadores, etc. [Koza, 92], [Kalganova, 00a].

En nuestro caso se simularán todos los circuitos lógicos secuenciales candidatos (implementación extrínseca) a partir de puertas lógicas nand y circuitos flip-flop tipo D.

- Implementaciones extrínsecas

En 1991 Louis y Rawlins diseñan circuitos lógicos combinatorios en donde cada uno de los genes de un individuo representa un tipo de compuerta con sus entradas [Louis, 91], [Louis, 93]. La evaluación de cada circuito se realiza mediante el mapeo a una matriz interconectada de nivel a nivel. La mayoría de los investigadores que diseñan circuitos extrínsecos al nivel de compuertas incorporan algunas mejoras al modelo de Louis.

Chen propone un diseño de circuitos a nivel de funciones empleando un modelo de grafos y Programación Genética que permite manejar circuitos de alta complejidad con un mecanismo de cruce que precisa de una gran potencia de cálculo [Chen, 02].

Coello y sus colegas han realizado implementaciones con Algoritmos Genéticos [Coello, 96, 97] y Colonia de Hormigas [Coello, 00] empleando un esquema en donde cada celda está representada por dos entradas y una puerta lógica que es mapeada posteriormente a una cadena cromosómica a la cual se le aplican los operadores genéticos. La evaluación de la aptitud de cada circuito se realiza en dos etapas, en la primera se busca obtener circuitos funcionales y en la segunda se bonifican los circuitos sintetizados en función de distintos criterios. Para Coello la reutilización de componentes lógicos cobra cierta importancia.

El trabajo de Miller muestra ciertas similitudes con el de Coello [Thomson, 98] que pasó de un enfoque inicial de nivel intrínseco [Miller, 97] a otro posterior extrínseco [Miller, 98]. El grupo de Miller representa en cada celda una cierta combinación de entradas y puerta lógica y la función de aptitud busca únicamente soluciones funcionales. Actualmente Miller y Thompson defienden los beneficios de partir de una solución funcional para obtener lo que denominan un camino neutral hacia la zona de diseño [Thomson, 98] y de la inyección de conocimiento para definir nuevas reglas tal como lo hace Islas en [Islas, 01].

Kalganova y Miller emplean una técnica de diseño de circuitos funcionales más compactos denominada erróneamente multiobjetivo pues tratan de minimizar una

solución eliminando componentes innecesarios [Kalganova, 99]. Este tipo de diseño en dos etapas es ampliamente utilizado en esta área. Kalganova, además, propone un diseño incremental que pretende diseñar sistemas complejos a partir de pequeños componentes que coevolucionan en varias poblaciones y donde el nivel de compuertas y funciones se combinan para lograr la evolución de los circuitos a partir de pequeños componentes reutilizables [Kalganova, 00b].

En 1992 J. Koza emplea en su libro *Genetic Programming* [Koza, 92] un esquema de programación genética con el objetivo de generar diseños funcionales empleando un encapsulamiento de subárboles en terminales para la reutilización de código o componentes.

- Implementaciones intrínsecas

Higuchi e Iba emplean representaciones matriciales y cromosomas de longitud variable en un intento de proporcionar cierto carácter de aprendizaje al diseño de circuitos a nivel de compuertas [Higuchi, 92, 96], [Iba, 97]. Este hardware se comporta como una matriz lógica programable y recibe una cadena de bits con instrucciones que lo configuran como un circuito determinado.

Thompson emplea Programación Genética y circuitos reconfigurables FPGA's (Field Programmable Gate Array) para diseñar circuitos digitales a nivel de compuertas y funciones mediante la reutilización de componentes [Thompson, 00]. Con estas técnicas se han producido quizás los mejores modelos generados hasta el momento.

- Enfoques multiobjetivo y distribuido

El objetivo principal de las técnicas multiobjetivo es la reducción de esfuerzo computacional mediante la reducción del número de evaluaciones de un circuito funcional [Coello, 00], [Rodríguez, 01], [Suto, 93]. Los mecanismos distribuidos y paralelos obtienen soluciones de mayor calidad y diversidad con menor esfuerzo. Ricardo Poli ha investigado en este terreno [Poli, 00].

Coello en [Coello, 00] reduce el esfuerzo computacional, evaluando por separado las 2n salidas independientes y la salida total del circuito generando un esquema similar a VEGA [Schaffer, 98].

E. de Jong emplea los conceptos de óptimo de Pareto en su método multiobjetivo para reducir considerablemente el esfuerzo computacional en circuitos tipo "even-n-parity" que presentan un espacio de soluciones especialmente difícil, teniendo en cuenta la funcionalidad, el tamaño y la diversidad de los árboles funcionales generados.

Al final de los años ochenta y principios de los noventa algunos autores, como Harvey [Harvey, 93], Cliff [Cliff, 93] y Beer y Gallagher [Beer, 92], propusieron la evolución artificial como un medio para automatizar el proceso de diseño de este tipo de procesos usando el concepto de Hardware Evolutivo introducido por Higuchi [Higuchi, 92, 96]. La idea básica de esta aproximación es decidir las relaciones de entrada/salida del sistema, proporcionando algunos bloques constructivos y permitiendo que un algoritmo genético o algún tipo de estrategia evolutiva propongan una composición de estos bloques para la optimización de su comportamiento.

Uno de los campos dónde se han aplicado extensamente estos conceptos es el correspondiente al diseño de circuitos electrónicos. Muchos autores basan la especificación del circuito electrónico requerido en una colección de pares de entrada/salida para controlar paramétricamente FPGAs (Field Programmable Gate Arrays) o algún otro tipo de estructura electrónica que desarrolle el circuito óptimo. Ejemplos de este tipo son los filtros digitales de Miller en [Miller, 97, 00], el ecualizador adaptativo de Murakawa [Murakawa, 96] o las metodologías de Lohn [Lohn, 95] y Torresen [Torresen, 02]. Sin embargo, aunque estas aproximaciones han tenido un éxito relativo, no tratan el problema principal del diseño automático de plantas industriales realmente complejas y es que, en general, en una planta real coexistirán varios sistemas electrónicos combinados con otro tipo de actuadores, sensores y pares entrada/salida. De hecho, es preferible que estos pares no hagan referencia a las variables relativas al circuito electrónico (los voltajes, corrientes, etc.) sino que deberían referirse a términos ambiguos como la productividad, los costes, las averías, los atascos, etc. El diseñador de la planta real trabaja al nivel de planta con conceptos como el incremento de productividad, minimización de las averías y atascos y costes.

Esto es lo que se pretende al incorporar en el diseño de plantas de manufactura la metodología evolutiva especificada en esta tesis para así poder diseñar combinaciones de sistemas de bajo nivel (circuitos electrónicos digitales) sin la especificación directa de sus relaciones entrada/salida teniendo en cuenta la planta y las restricciones de alto nivel impuestas al sistema en general.

7.2.2.6 Implementación de la metodología evolutiva de diseño automático

La idea básica es introducir en el sistema evolutivo un simulador de procesos discretos que evalúe el comportamiento de la planta bajo distintas condiciones de carga en función de restricciones de alto nivel (productividad, el coste de energía, etc.). Esta aproximación en el caso de plantas complejas consume de muchos recursos y su distribución en un cluster de ordenadores disminuye en gran medida los tiempos de ejecución hasta hacerlos razonables.

En la figura se ilustra la disposición de los sensores en el tramo de carga y descarga del horno.

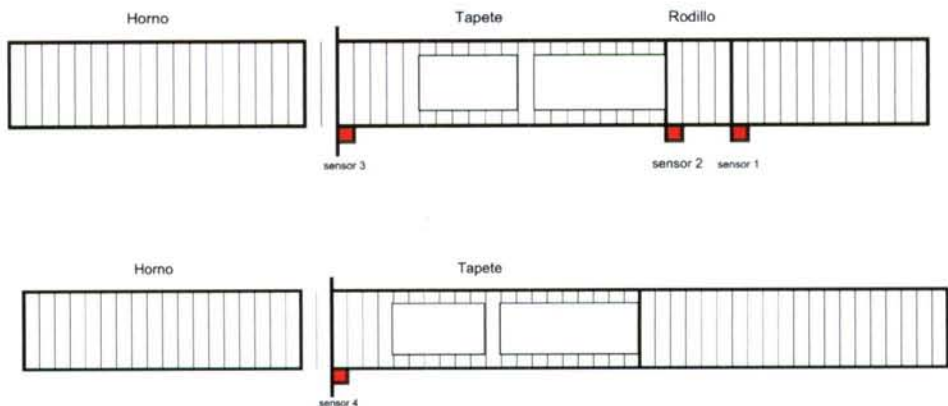


Figura 7.35: Disposición de los sensores en la línea

En la figura se muestran las entradas (sensor1, sensor2, sensor3 y sensor4) y salidas del macro-controlador.

VRe velocidad del rodillo de entrada,
 VTe velocidad del tapete de entrada,
 VTs velocidad del tapete de salida.



Figura 7.36: Entradas y salidas del controlador

Es necesario indicar que no existe restricción de ningún tipo en las entradas y salidas de los controladores. Es decir, puede ocurrir que un controlador haga uso de todos los valores de los sensores y de un máximo de 10 nodos, o puede ocurrir el caso opuesto donde directamente un solo sensor esté conectado directamente a una de las salidas.

Es de esperar que casi todo el peso del controlador recaiga en el control del tapete de salida pues como se vio en la realidad y según los resultados obtenidos del estudio realizado para la planta real, éste es el elemento crítico de la planta.

- Codificación

En este caso el cromosoma es de longitud variable y está formado por tres subcromosomas que codifican los tres circuitos de control existentes. La longitud máxima de cada subcromosoma es de 30 genes. Para evitar tratar con matrices del mismo tamaño y dispersas, se crea una única matriz sin ceros y donde se incorporan los elementos necesarios para indicar la posición de cada elemento en la matriz inicial.

Se adoptó desde un principio la codificación en forma de árbol binario dado que todos y cada uno de los nodos que intervienen en el circuito (puertas NAND y circuitos FLIP-FLOP) presentan dos entradas y que el circuito en general parte de varias entradas hasta confluir en una única salida.

Cada entrada, salida y puerta lógica es codificada de tal modo que la estructura del cromosoma es del tipo de árbol binario. Los nodos del árbol corresponden a las puertas nand (codificados como un 5 en el cromosoma) y circuitos D flip-flop (codificados

como un 6). Las hojas del árbol binario (codificados como 1, 2, 3 o 4) son los sensores de la línea.

- Función de calidad

La función que evalúa la calidad de cada uno de los individuos es la suma ponderada de tres factores en donde se incluye:

- La energía consumida por los motores que controlan los transportes de entrada y salida del horno.
- Las esperas:
 - en la descarga del horno cuando está ocupado el transporte de salida, y
 - en la carga del horno cuando está ocupado el transporte de entrada.
- La productividad de la línea.

La energía consumida es calculada como el valor medio de la diferencia entre el valor de las velocidades y su valor mínimo.

$$\text{energía} = \text{media} (V_{re} - V_{remin} * 6.8) + \text{media} (V_{te} - V_{temin} * 6.8) + \text{media} (V_{ts} - V_{tsmin} * 10)$$

La productividad máxima es de 111 tableros (para el tiempo de simulación definido) y la relación de pesos entre los términos de la ecuación general es de 60%, 30% 10%.

$$\text{sensor} = \left(\frac{\text{productividad}}{111} \right) * \left(60 + \frac{30}{\text{retrasos} + 1} + \frac{10}{\text{energía} + 1} \right)$$

$$\text{calidad} = \left(\frac{1}{\text{sensor} + 1} \right)$$

Existe un factor adicional que interviene en la calidad de los individuos de modo inversamente proporcional a la longitud de su cromosoma que será aplicado después de algunas simulaciones. Luego veremos cómo.

- Operadores genéticos

Los operadores genéticos que intervienen en la evolución son:

- Selección por torneo con ventana de dos individuos;
- Mutación (la probabilidad de cada uno de estos operadores es de un 1%).
 - Creación de un subárbol.
Substitución de una hoja por un subárbol formado por un nodo y dos hojas de valores aleatorios.
 - Modificación de nodos y hojas del árbol.
Cambio de valor de la codificación de un nodo (4 o 5) y/o hoja (1, 2, 3 o 4).
 - Eliminación de un subárbol.
Substitución de un nodo terminal (conectado a dos hojas) por una hoja de valor aleatorio.

- Cruce

A continuación se muestra un ejemplo del operador cruce aplicado a este tipo de codificaciones:

Cromosoma1 = 5 4 6 2 1

Cromosoma2 = 6 4 5 3 1

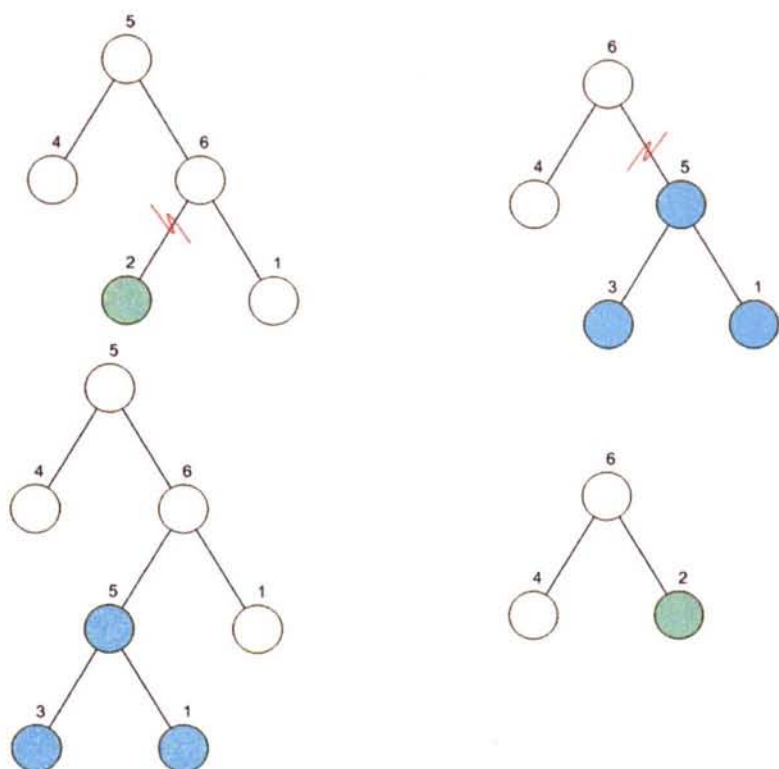


Figura 7.37: Cruce con codificación en árbol

- Descripción de la evolución

La tabla siguiente describe el valor de los parámetros básicos que intervienen en la función de calidad y su evolución.

Notación:

Gnr = Generación

Ind = Individuo

Prd = Productividad

Ad = Atascos descarga

Ac = Atascos carga

Ere = Energía rodillo entrada

Ete = Energía tapete entrada

Ers = Energía rodillo salida

nand = Puertas NAND

fp = Circuitos FLIP-FLOP

Gnr	Ind	Prd	Ad	Ac	Ere	Ete	Ers	nand	fp
50	1	174	0	4.5	5.47	4.26	6	6	27
50	100	0	1	0.81	5.23	7.3	3	8	27
50	200	2	0	4.5	1.13	14.78	4	8	27
50	300	0	0	0.81	2.13	4.85	2	10	27
120	1	0	0	4.5	1.13	4.05	2	1	9
120	100	0	0	23.16	1.13	5.4	5	4	21
120	200	0	0	4.5	1.13	4.06	3	3	19
120	300	0	0	4.5	1.13	5.4	0	1	50
140	1	4	1	2.59	2.59	4.11	1	2	9
140	100	0	0	4.5	1.13	4.06	1	2	9
140	200	0	0	4.5	1.13	1.97	4	1	5
140	300	180	0	4.5	1.13	3.47	0	1	13
250	1	0	0	4.5	1.13	5.4	0	1	5
250	100	0	0	4.5	1.13	1.93	4	1	5
250	200	0	0	4.5	1.13	5.4	0	2	7
250	300	0	0	4.5	1.13	5.4	0	1	13
350	1	0	0	4.5	1.13	5.4	0	1	5
350	100	0	0	4.5	23.15	4.85	1	1	7
350	200	0	0	4.5	1.13	1.93	4	1	5
350	300	0	0	4.5	1.13	5.4	0	1	13

Lo indicado en la tabla se muestra en las gráficas correspondientes al valor máximo y medio de las calidades y número de bloques de cada generación que se presentan en las figuras 7.38 y 7.39 respectivamente. En este caso, los individuos que integran cada generación son 300, dado que la máxima cantidad de elementos que constituyen el árbol son 30 en total (10 por cada circuito).

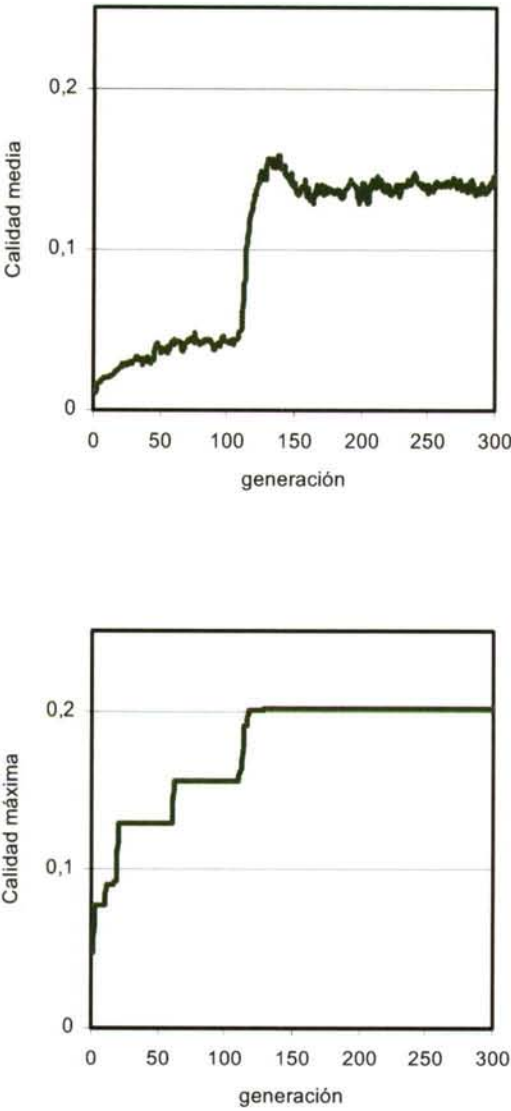


Figura 7.38: Valor medio y máximo de la calidad y número de bloques

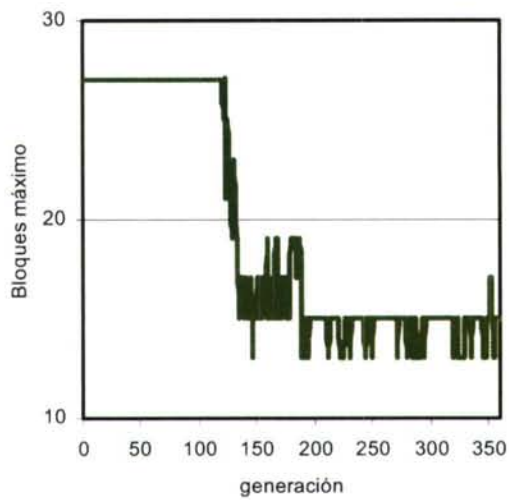
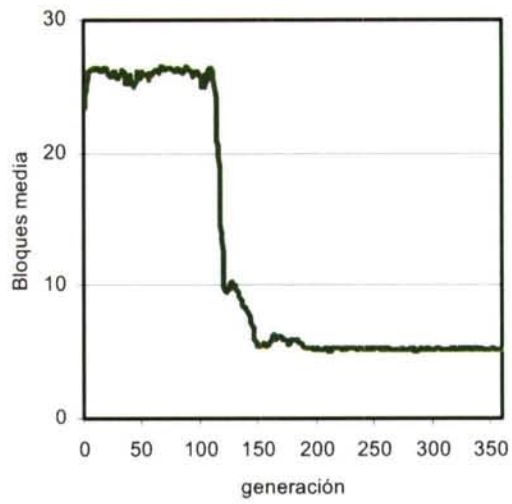


Figura 7.39: Valor medio y máximo de la calidad y número de bloques

A la vista de estos resultados es fácil ver que la evolución se divide en tres tramos o fases:

- Fase 1

Aplicación de la función de calidad básica:

$$sensor = \left(\frac{productividad}{111} \right) \times \left(60 + \frac{30}{retrasos + 1} + \frac{10}{energía + 1} \right)$$

$$calidad = \left(\frac{1}{sensor + 1} \right)$$

- Fase 2

Aplicación de un factor de corrección a partir de la generación 111 consistente en asignar mayor calidad a aquellas soluciones con el menor número de bloques.

Se consideran como bloques no solamente las puertas lógicas sino también el número de señales que forman parte del circuito, es decir, el número total de elementos que forman el árbol cromosómico.

La función de calidad pasa a ser la siguiente:

$$sensor = \left(\frac{productividad}{111} \right) \times \left(60 + \frac{30}{retrasos + 1} + \frac{10}{energía + 1} \right)$$

$$calidad = \left(\frac{1}{sensor + 1} \right) + (27 - bloques) \times 0.005$$

Se observa como en este tramo y hasta el comienzo de la fase3 (generación 140) se produce un ascenso acusado de la calidad media desde un valor 0.05 hasta 0.15 debido a que gracias a la incorporación de este factor se elimina considerablemente el número elementos que intervienen en el controlador. Al mismo tiempo se observa como el número de bloques presenta un fuerte descenso y pasa a ser 7 partiendo de un valor inicial próximo a 27.

- Fase 3

Una vez observada una estabilización en la calidad máxima de la población (generación 140), se aplicó un operador que permite mutaciones consistentes en la eliminación de subramas del árbol cromosómico. La probabilidad de este operador específico es de 0.01

En la gráfica correspondiente se observa cómo la calidad media presenta un pequeño descenso debido a la aparición de este operador aunque la calidad máxima no se ve afectada. Al mismo tiempo desciende el número de bloques del circuito.

De este modo, en las primeras generaciones la población de controladores evolucionará hacia controladores óptimos sin ninguna limitación en su longitud y a partir de la aplicación de este operador se intentará reducir su tamaño al máximo sin menoscabo de su calidad. El motivo de la aplicación de este criterio es la prioridad del rendimiento del controlador frente al coste del mismo pues en este caso los elementos del circuito no son caros (no ocurría lo mismo cuando eran máquinas de una línea).

A continuación se realiza una comparativa entre una solución intermedia (generación 111) y la mejor de las obtenidas (generación 360) según la función de calidad descrita.

• Circuito de calidad intermedia

Gnr	Ind	Prd	Ad	Ac	Ere	Ete	Ers	nand	fp
111	300	111	0	0	0.815	2.13	3.49	8	4

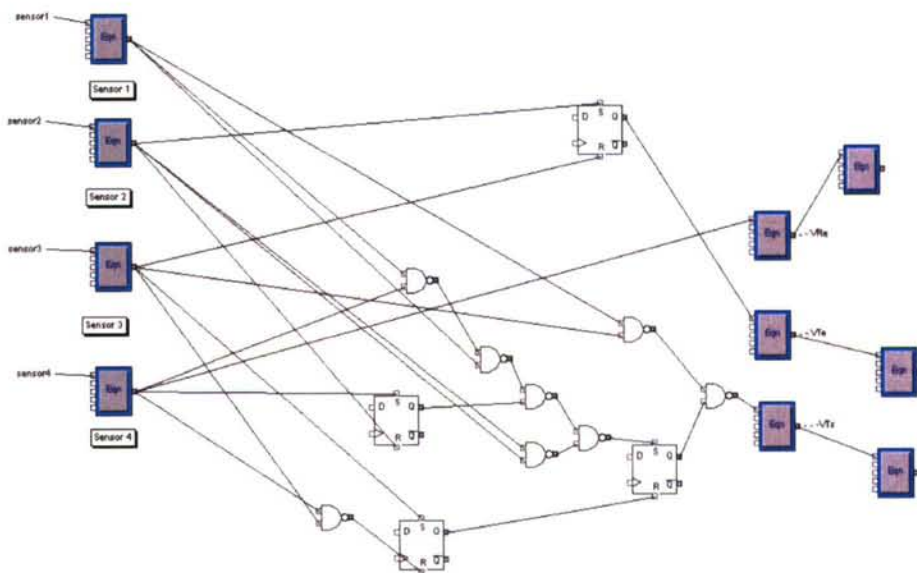


Figura 7.40: Circuito de calidad intermedia

Este individuo de la generación 111, es decir, justo al término de la fase1, presenta tres circuitos con las siguientes características:

- Control de velocidad del rodillo de entrada
La señal de control es la propia señal del sensor 4.
- Control de velocidad del tapete de entrada
La señal de control es la salida un flip-flop cuyas entradas son el sensor 2 y 3.
- Control de velocidad del tapete de salida
La señal de control es la salida de un circuito de 8 puertas nand y 3 flip-flop.

Como muestra del control de velocidad del tapete de salida (circuito 3) se presenta la siguiente gráfica. En ella podemos observar cómo el tapete trabaja inicialmente a una velocidad constante máxima hasta que se detecta la presencia de tableros en el tapete y

comienza el control. A partir de esta generación se incluirá en la función de calidad un término que da prioridad a los circuitos más pequeños.

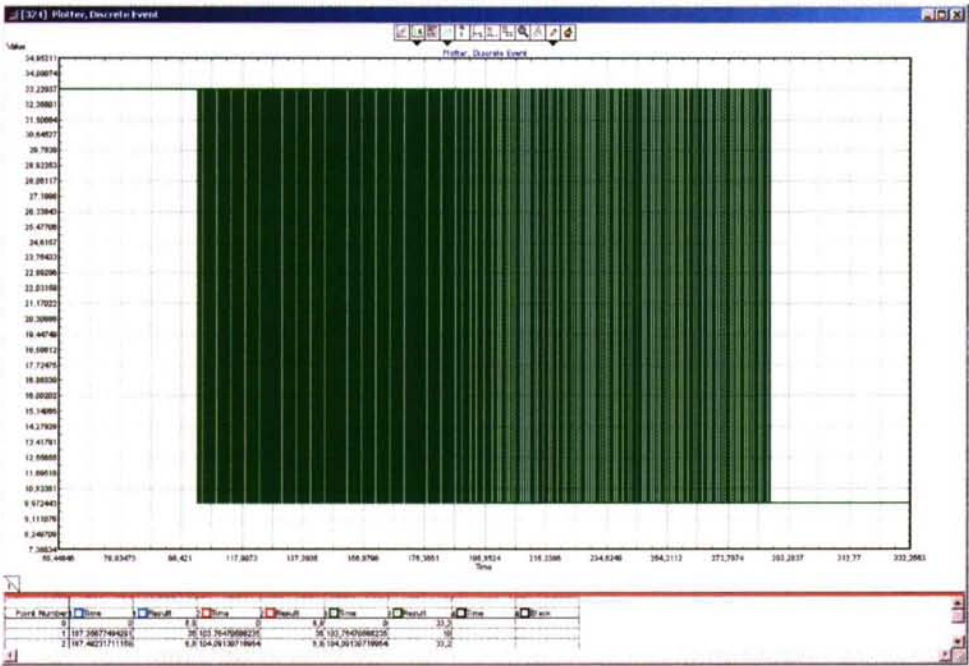


Figura 7.41: Control de velocidad del tapete de salida

- Circuito solución

Gnr	Ind	Prd	Ad	Ac	Ere	Ete	Ers	nand	fp
360	300	111	0	0	4.5	1.13	5.4	1	0

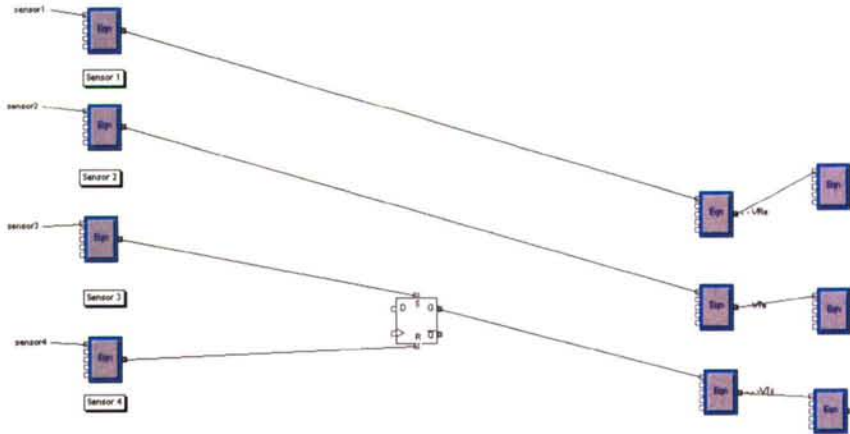


Figura 7.42: Circuito de mayor calidad

El individuo de la última generación (generación 360) presenta tres circuitos con las siguientes características:

- Control de velocidad del rodillo de entrada
La señal de control es la señal del sensor 1.
- Control de velocidad del tapete de entrada
La señal de control es la señal del sensor 2.
- Control de velocidad del tapete de salida
La señal de control es la salida de un circuito flip-flop cuyas entradas son los sensores 2 y 3.

El control de velocidad del tapete de salida (circuito 3) se muestra en la siguiente gráfica donde el tapete trabaja inicialmente a velocidad máxima y al ir saliendo los tapetes comienza el control. Es decir, el comportamiento de la lógica de control es similar al caso anterior pero en este caso el circuito está formado por muchas menos puertas al haberse aplicado anteriormente el operador de eliminación.

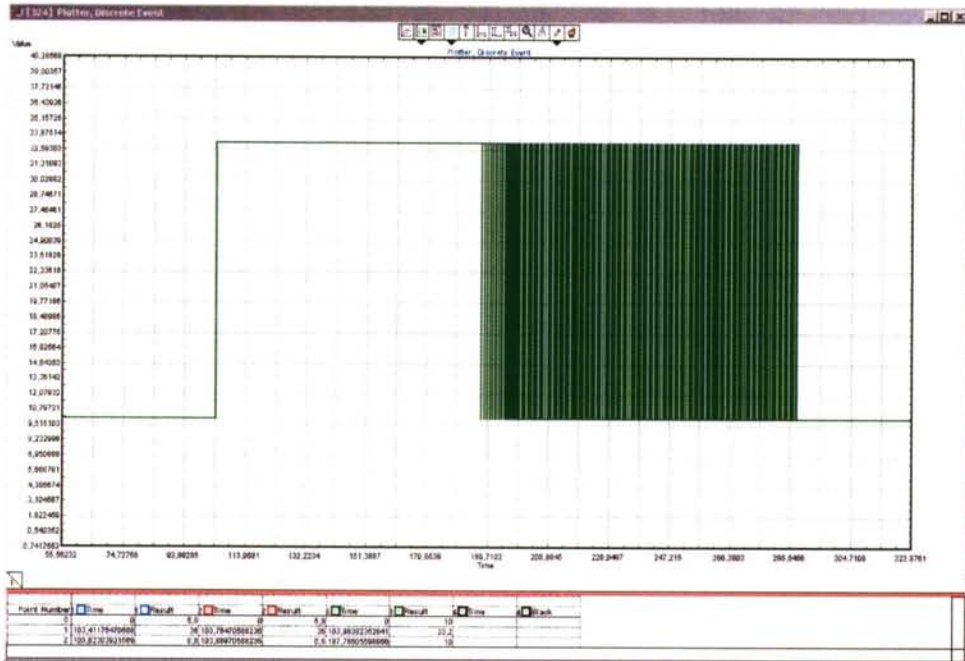


Figura 7.43: Control de velocidad del tapete de salida

A continuación se muestra con más detalle el control del tapete de salida. Se observa como al inicio los tableros son de 4.5 metros (escalones de velocidad más anchos) y después de 0.5 (más estrechos).

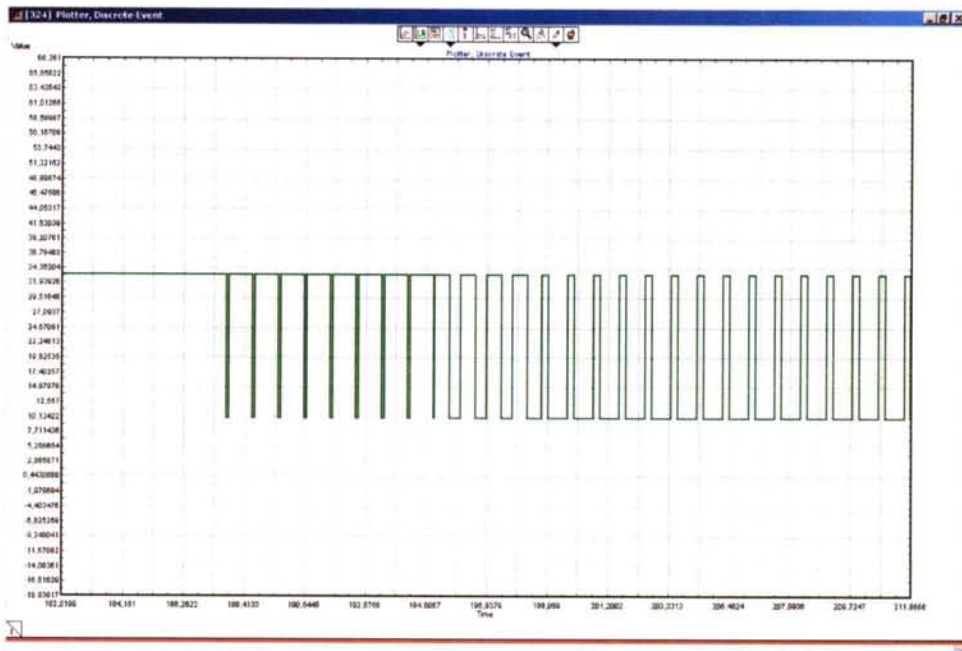


Figura 7.44: Detalle del control de velocidad del tapete de salida

En el control del tapete y rodillo de entrada se observa que cuando el tablero es de 4.5 metros las curvas que representan las velocidades de los transportes se solapan, pero cuando los tableros son de 0.5, dejan de hacerlo.

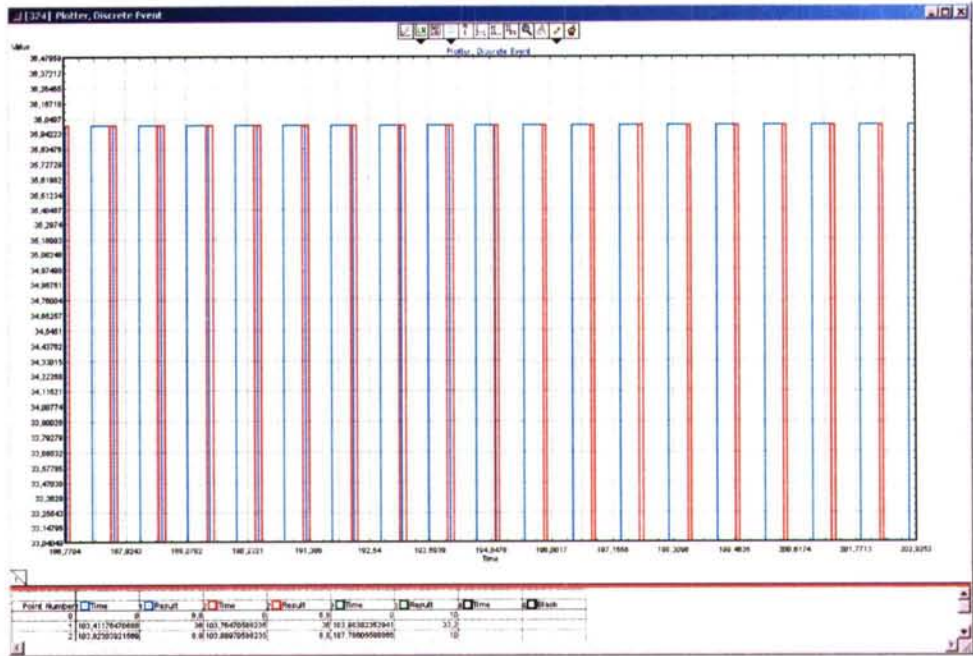


Figura 7.45: Control de velocidad del tapete y rodillo de entrada

7.2.3 Conclusiones del diseño de controladores

En este capítulo se aplicó la versatilidad del entorno de diseño expuesto para la optimización de la disposición y control de las máquinas que forman parte de dos líneas de producción industrial. Para ello, se optimizaron parámetros de carácter general como la productividad que, aunque están muy cerca del dominio del usuario, interviene indirectamente en el diseño de cada controlador. En este caso, el problema de optimización de la disposición de maquinaria se aplica a una línea de ensamblaje ficticia de dos subproductos complementarios que presentan cadencias distintas y el correspondiente a la síntesis de controladores secuenciales lo hace en una línea de barnizado real de la empresa SANTOS Cocinas y Baños.

Los resultados obtenidos en el primero de los casos muestran el entorno de diseño desarrollado como una herramienta completamente válida para el diseño de plantas industriales optimizando el tipo y cantidad de máquinas necesarias en función del objetivo requerido. La posibilidad de implantar en el simulador comercial que se dispone (Extend) toda una librería de bloques que definen el comportamiento de gran variedad de recursos nos permite que, en combinación con el resto de los módulos que forman parte del entorno de diseño, sea posible emular el comportamiento de una línea real y obtener, según los preceptos indicados, la disposición y tipo de maquinaria que más se ajuste a los requerimientos y especificaciones definidas. De este modo, es posible determinar de modo automático y definiendo las especificaciones en términos de alto nivel (productividad, número de máquinas y pérdidas) cual será la combinación de máquinas que más se ajuste al funcionamiento deseado de la línea.

En el caso de la planta de barnizado real, se optimizó su productividad minimizando el gasto energético del proceso mediante la síntesis de circuitos lógicos secuenciales basados en puertas lógicas y relojes codificados en forma de árbol binario. Lo más destacado en esta implementación es la versatilidad del sistema a la hora de definir codificaciones de todo tipo. Las modificaciones realizadas en este sentido (mapeado y operadores genéticos) fueron mínimas y demostraron los beneficios de la modularidad del entorno. Además se demostró cómo es posible diseñar automáticamente y al mismo tiempo tres circuitos secuenciales para el control de un sistema de alta dificultad donde únicamente se especifican conceptos generales que evitan la necesidad de tener conocimientos específicos sobre el tema. Es de destacar que esta última implementación resultó enteramente satisfactoria y está actualmente funcionando en la planta real.

Diseño Automático de Morfologías 3D

8 Diseño Automático de Morfologías 3D

Hasta ahora se ha pretendido liberar al humano de la toma de decisiones que se realiza en el desarrollo de todo proceso de diseño haciendo uso del entorno de diseño automático desarrollado en esta tesis.

En todas las implementaciones realizadas hasta este punto se ha desligado al diseñador del proceso debido a sus limitaciones en el caso de tratar con problemas complejos, multidimensionales y con acoplamientos no lineales. Sin embargo, existen problemas en donde intervienen elementos subjetivos que necesitan de la colaboración del humano. En este tipo de diseño el humano interpreta el papel de usuario, un rol muy diferente al de diseñador al que nos tenía acostumbrados en el diseño tradicional.

En este capítulo se pretende demostrar la versatilidad de la metodología de diseño desarrollada incorporando al mismo un módulo interactivo para la evaluación múltiple y subjetiva de productos. Se aplicará este concepto para la generación interactiva mediante algoritmos genéticos de morfologías tridimensionales.

8.1 Introducción

Los algoritmos evolutivos se han convertido en una herramienta útil a la hora de afrontar problemas asociados a la optimización de productos, procesos y sistemas debido a sus características de convergencia, sencillez y robustez. Sin embargo, se les ha conferido poca atención para su aplicación en el diseño interactivo de productos.

La metodología evolutiva de diseño automático desarrollada en esta tesis y los clásicos sistemas CAD, que aunque han evolucionado no dejan de ser “sistemas computacionales de ayuda al diseñador”, sirven de base para el desarrollo de un nuevo “sistema computacional de ayuda al diseño” de carácter interactivo y evolutivo.

Esto es así porque, en el entorno definido, el software CAD deja de ser un sistema de diseño interactivo para formar parte del módulo de simulación. Para ello es necesario inhabilitar su interfaz clásica y automatizar la interpretación y representación tridimensional de la codificación de la población de morfologías que se pretende evolucionar. A partir de aquí se define una interfaz interactiva para el entorno de diseño que, haciendo uso de los modelos 3D generados, permite que una serie de usuarios evalúen su calidad.

8.2 Marco teórico

Llegados a este punto, se realizará a continuación una breve descripción de la evolución del diseño gráfico desde los sistemas CAD clásicos hasta el diseño interactivo-evolutivo que es donde se enmarca la aplicabilidad del entorno de diseño expuesto.

8.2.1 CAD y el proceso de diseño

El término diseño en el CAD clásico se utiliza para referirse a “la representación gráfica, de acuerdo con una idea creativa previa, de un objeto artístico o funcional, de un dispositivo mecánico o de la estructura o funcionamiento de un sistema o proceso”.

Según esta definición, las aplicaciones CAD clásicas inciden tan sólo en la representación computacional gráfica interactiva y se quedan cortas a la hora de pretender ser definidas como “aplicaciones de la informática al proceso de diseño” [Salmon, 87] y mucho menos si pretendemos hacer uso de ellas para su automatización.

Para llenar este hueco se implementaron nuevos sistemas CAD integrables con otro tipo de software en un intento de complementar el CAD clásico en la tarea de englobar en un único sistema computacional la totalidad del proceso de diseño. Aunque el objetivo no se alcanzó, se dio un paso adelante en este sentido.

Otro avance fue pretender diseñar modelos de forma automática aplicando estos nuevos sistemas CAD. Con este propósito se plantearon soluciones CAD que incorporaban algoritmos evolutivos para la optimización de diseños en función de la calidad asignada por un simulador que conjuntamente con el sistema CAD conformaban el sistema. Sin embargo estos entornos en la mayoría de los casos no son versátiles y pocas veces permiten evaluaciones subjetivas.

Lo que se pretende en esta tesis es incorporar dentro de esa función de calidad múltiple la subjetividad de un conjunto de usuarios. Siguiendo esta pauta se obtendría un sistema de diseño automático modular donde fuese posible incorporar a las evaluaciones técnicas obtenidas por simuladores, calidades subjetivas provenientes de una interfaz.

8.2.2 Evolución histórica del diseño gráfico

8.2.2.1 CAD clásico

El CAD ha constituido un hito para el sector de la ingeniería especialmente porque eliminó la necesidad de dibujar los planos a mano, permitiendo además incorporar los cambios con facilidad. Desde la aparición del modelado alámbrico a principios de la década de los 70 hasta la actualidad, en que se está investigando el diseño basado en Inteligencia Artificial, los sistemas CAD han ido evolucionando paralelamente al incremento de potencia de cálculo de los ordenadores y al consecuente avance en las prestaciones del software CAD asociado al proceso de diseño. También han influido en gran medida a este avance la miniaturización de los equipos, el desarrollo de las redes informáticas, la mayor accesibilidad económica del PC y del software CAD y las mejoras existentes en la usabilidad de los mismos. Además, la generación de estándares gráficos y la creación de nuevos interfaces orientados al usuario (realidad virtual y aumentada) han sido clave en esta evolución hasta el punto que el CAD ha alterado la naturaleza, definición y alcance del proceso de diseño [Baldasano, 89], [de Cos, 98].

Los sistemas CAD actuales cada vez más integrados han provocado un cambio fundamental en la planificación, gestión y comunicación de la información entre las distintas fases del proceso de diseño, mejorando así la productividad. Gracias a estos nuevos sistemas de diseño se ha avanzado mucho en las fases de concepción, detalle y

análisis, se han mejorando los resultados y se han acortado tiempos. Tanto es así que los sistemas CAD ocupan hoy día un lugar destacado en las empresas de ingeniería.

El concepto de Dibujo Asistido por Ordenador nace en la década de los 50 cuando el ejército norteamericano desarrolla los primeros plotters capaces de representar dibujos realizados por ordenador. Al mismo tiempo el MIT (Massachusetts Institute of Technology) presenta el SKETCHPAD, primer software de CAD a base de píxeles. Este mismo instituto en colaboración con las Fuerzas Aéreas de USA diseñó el SAGE (Semi Automatic Ground Environment), primer sistema de gráficos por ordenador que permitía la visualización de datos provenientes de un radar.

En 1960 se mencionó por primera vez el término “Diseño asistido por ordenador” en un anteproyecto del MIT, titulado “Computer-Aided Design Project” y dirigido por Douglas Ross y Dwight Baumann [Ross, 93]. Ocho años más tarde ya estaban disponibles los primeros sistemas CAD 2D que necesitaban de grandes ordenadores (mainframes) y que fueron implantados masivamente en la industria automovilística, naval y aeronáutica (General Motors y Bell Telephones). En los años siguientes se inventó el lápiz óptico, se realizaron las primeras aplicaciones CAM y se perfeccionó el software gráfico.

El problema del diseño de superficies se resolvió inicialmente utilizando curvas y superficies fácilmente representables (círculos, rectas, cilindros, conos, etc.). Los diseños más complicados como los cascos de buques (Steve Coons [Coons, 64]), el fuselaje y las alas de aviones (J. Fergusson en Boeing 1964 [Fergusson, 64]) o las carrocerías de coches (Paul de Castelju en Citroën [Castelju, 58] y Pierre Bézier en Renault [Bézier, 58]) seguían procesos más sofisticados.

Los primeros antecedentes en el modelado de sólidos son los trabajos desarrollados por Coons [Coons, 1967] en el MIT en los 60 que se centraron en la aplicación de métodos numéricos para generar sólidos por barrido. Sin embargo fueron Aristides Requicha y Herbert Voelcker de la Universidad de Rochester quienes sentaron las bases del desarrollo del modelado de sólidos como disciplina. Otro investigador importante en este campo fue Baumgart que trabajó en 1974 en la representación mediante aristas aladas y en la edición con operadores de Euler. Como resultado todos estos trabajos de investigación surgieron a finales de los años 60 y principios de los 70 las primeras aplicaciones para el modelado de sólidos. Entre ellos destacan EUCLID de J.M. Brun, PADL-1 de la Universidad de Rochester, Shapes del MIT y TIPS-1 de Okino.

A mediados de la década de los 90 aparecen en el mercado numerosos programas de CAD tridimensionales de distinto grado de complejidad y de gran variedad de usos y aplicaciones. Durante esta década surgen soluciones específicas a segmentos verticales tales como la construcción, ingeniería civil, mecánica, fabricación, etc. A partir de entonces y hasta la actualidad han ido apareciendo paulatinamente el diseño paramétrico, el diseño variacional y la asociatividad completa. Al mismo tiempo aparece en estos años los primeros conceptos de interactividad del software de diseño. La realidad virtual (RV) y la aumentada (RA) son dos tecnologías en máxima expansión que representan dos líneas de investigación en pleno desarrollo.

La realidad virtual es un tipo de simulación implícitamente interactiva de un escenario tridimensional que responde, en tiempo real, a los estímulos del usuario. La revolución de la RV respecto a la interacción hombre-máquina reside en que el usuario deja de percibir el ordenador para pasar a interactuar directamente con la escena. La realidad

aumentada, sin embargo, es una tecnología en la que la visión que tiene el usuario del mundo real es mejorada o aumentada con información adicional generada por un modelo informatizado. Los sistemas RA llevan el ordenador al mundo real del usuario, mientras que los sistemas RV llevan el mundo real al ordenador.

Una vez esbozada la historia del CAD clásico se plantean a continuación la evolución del mismo hacia la integración con otras herramientas que participan en el proceso de diseño y se abunda un poco más en el desarrollo de su interactividad.

8.2.2.2 Integración vertical de aplicaciones de diseño

En paralelo al desarrollo de las aplicaciones de diseño asistido por ordenador, se han desarrollado también otras de simulación, modelización y manufactura de productos, al mismo tiempo que se producía un aumento de la capacidad de proceso y de la facilidad de uso de los PCs. La combinación de este tipo de aplicaciones dio lugar a la aparición del concepto de “integración vertical” en el diseño de productos y procesos.

En el pasado esto resultaba imposible debido a la inexistencia de un estándar para la exportación de datos de una aplicación a otra, pero actualmente el CAD se encuentra totalmente integrado con aplicaciones CAM y CAE. Esta integración permite realizar automáticamente el dibujo de detalle y la documentación del diseño y posibilita la utilización de métodos numéricos para realizar simulaciones sobre el modelo como una alternativa a la construcción de prototipos. La aplicación de este nuevo concepto supone un ahorro importante debido a que la duración del proceso de diseño se reduce al mejorar la detección de errores y el flujo de información entre aplicaciones de distinto tipo. Esto posibilita la síntesis de un entorno de colaboración que más tarde dio origen al concepto de Ingeniería Concurrente.

Hoy en día, casi todas las aplicaciones CAD que encontramos en el mercado parten de objetos tridimensionales mediante los cuales se pueden realizar proyecciones bidimensionales, descripciones volumétricas, ensamblajes, cálculo de elementos finitos, simulación de programas CNC (Computer Numerical Controls), exportación de ficheros “STL” (Stereo Lithography) para máquinas de prototipado rápido o incluso poseen bases de datos que almacenan sus propiedades y características (materiales, acabados, tolerancias, etc.). Con esta integración, tan sólo las etapas de definición y ensayo con prototipos quedan fuera del ámbito del sistema CAD. Sin embargo, la importancia de la realización de ensayos con prototipos dependerá de la naturaleza del modelo a diseñar y de la posibilidad de sustituirlos por simulaciones numéricas.

La implementación de la integración vertical de aplicaciones puede seguir dos vías diferenciadas: las aplicaciones llaves en mano o las interfases genéricas.

Las aplicaciones “llave en mano” consisten en aplicaciones estrechamente integradas, en donde el análisis por elementos finitos, el diseño e incluso la gestión de la base de datos de ingeniería se añaden a un CAD concreto, con el fin de crear un conjunto de herramientas de diseño. Este tipo de aplicaciones tiene un cierto número de ventajas, ya que suelen ir dirigidas a un segmento muy concreto, pero presentan ciertas desventajas en cuanto se intentan utilizar fuera de aquel contexto concreto para el que fueron desarrolladas. Por otro lado, las aplicaciones específicas generadas por empresas de CAD raramente alcanzan los niveles de calidad de aquellas desarrolladas con un objetivo específico.

Actualmente la mayoría de aplicaciones utilizan estándares como IGES o STEP para intercambio de datos entre aplicaciones de diferentes empresas de software. Lamentablemente, la importación de datos desde una aplicación a otra nunca es totalmente satisfactoria, perdiéndose parte de la información contenida en el modelo original. Esta nueva aproximación permite que con la integración vertical se obtenga beneficio de aquellas aplicaciones CAD que han tenido mayor éxito en el mercado y se integren mediante APIs (Application Program Interfaces) en el entorno de trabajo desarrollado. Con la integración a nivel de APIs, los productos que se integran al CAD contienen toda la riqueza de detalle de las aplicaciones individuales, pero se hallan integradas al CAD.

8.2.2.3 Diseño interactivo.

Al mismo tiempo que se concentraban esfuerzos en integrar el CAD clásico con otro tipo de aplicaciones se estudió la posibilidad de mejorar su interacción con el usuario en un intento de hacerlo más partícipe del diseño.

Walter Benjamin [Benjamin, 73] ya planteaba, a principios del siglo XX, la problemática de la interactividad con el diseño. En su opinión la distinción entre diseñador y usuario empezaba a romperse y éste último comenzaba a participar directamente en la creación del diseño.

A continuación se hará una breve descripción de los conceptos fundamentales del Diseño Interactivo y de su evolución histórica.

8.2.2.3.1 Interfases de usuario

Jones [Jones, 81] definió el Diseño en Ingeniería en 1981 como el “empleo de principios científicos, información técnica e imaginación para definir máquinas, estructuras mecánicas o sistemas que ejecuten unas funciones especificadas previamente, con el máximo de economía y eficiencia”.

Según esta definición el proceso de diseño de interfaces de usuario presenta características comunes con otros procesos de diseño de sistemas. Sin embargo, se diferencia de estos últimos en que está centrado en los usuarios, presenta un alto componente artístico y psicológico y necesita de la ayuda de usuarios finales para comprobar la usabilidad de los interfaces creados.

El objetivo principal de toda interfaz de usuario es el desarrollo de la interacción del mismo con el ordenador, garantizando su utilidad, seguridad, eficiencia y facilidad de uso y aprendizaje. La interfaz se convierte de este modo en una superficie de contacto [Laurel, 90] hombre-máquina a través de la cual el usuario se comunica con el sistema. Tomando las palabras de Negroponte una interfaz es “el sitio donde se encuentran los bits y las personas” [Negroponte, 94].

Para que un sistema interactivo cumpla sus objetivos tiene que ser lógico, intuitivo, consistente, robusto frente a los errores, usable y accesible a gran variedad de usuarios. Tal es su importancia que una pobre implementación reduce la funcionalidad de un sistema, aumenta el tiempo de aprendizaje y origina niveles de errores inaceptables.

Toda interfaz representa la puerta del usuario a la funcionalidad de un sistema [Nielsen, 93]. En este sentido la usabilidad de una interfaz hace referencia a la medida en que le

es útil al usuario para conseguir sus objetivos específicos con efectividad, eficiencia y satisfacción. Los interfaces de usuario usables hacen que el software sea fácil de utilizar y aprender, se reduzcan los costes de producción, mantenimiento y uso y se mejore la calidad del producto.

Debido a la importancia de la usabilidad la interfaz ésta debe estar basada en términos y conceptos propios del usuario en lugar de en conceptos basados en la aplicación. Una interfaz debe ser evaluada en función de la rapidez de aprendizaje, la velocidad de operación, su robustez (tolerancia frente a los errores del usuario) y su adaptabilidad a diferentes modelos de trabajo. De este modo, el usuario es involucrado en el proceso de diseño: "No quiero concentrar mis energías en la interfaz, me quiero concentrar en mi trabajo" [Norman, 88].

8.2.2.3.2 Evolución de interfaces de usuarios

Aunque la variedad de estilos de interacción es alta (manipulación directa, sistemas de menú, formularios, lenguajes de comando, lenguajes naturales, etc.) , la mayor parte de los usuarios interaccionan con los sistemas usando interfaces gráficas tipo menú debido a su facilidad de aprendizaje. Este tipo de interacción es el resultado de una evolución a lo largo de cinco generaciones.

La primera generación de interfaces de usuario la constituyen los primeros interfaces interactivos llegados con la aparición de los CRTs. Este tipo de interfases se basa en la introducción de instrucciones con formato fijo. Más adelante con la aparición de la segunda generación se introdujeron los menús simples (donde el usuario introduce el código asociado) y las primeras herramientas de carácter gráfico (ratón).

La tercera generación surgió con la aparición de sistemas gráficos de ventanas, iconos, menús desplegables y elementos apuntadores. En estos interfaces se visualizan diferentes tipos de información simultáneamente y se reduce el uso del teclado para la especificación de órdenes.

Cuando se implementa el hipertexto y la multitarea nace la cuarta generación de interfaces. Por último la quinta generación está asociada a las nuevas tendencias (hipermedia) y su objetivo final es la Manipulación Directa.

Como resultado de la evolución de los interfaces de usuario aparece el Diseño Interactivo-Evolutivo. En este nuevo concepto de diseño automático el usuario marca las pautas del desarrollo del producto final mediante la evaluación directa de soluciones intermedias a través de una interfaz. A continuación se expone su descripción básica.

8.2.2.4 Diseño Interactivo Evolutivo

El Diseño Interactivo Evolutivo se aplica en sistemas de optimización que dependen de la evaluación humana parcial o completa de la aptitud de las soluciones generadas mediante técnicas de búsqueda evolutiva. Su objetivo consiste en lograr una sinergia hombre-máquina que logre una mayor creatividad y efectividad que la que el ser humano o los Sistemas Evolutivos logren de forma independiente. Así, esta participación de los sistemas de evolución artificial en el proceso de diseño no consiste únicamente en liberar al usuario de tareas mecánicas difíciles y repetitivas sino que al mismo tiempo pretende sacar el mayor partido posible a la creatividad del ser humano en el proceso de diseño.

En Diseño Interactivo Evolutivo los usuarios en su papel de observadores externos influyen de forma interactiva y transparente en el proceso de diseño seleccionando, por ejemplo, las imágenes de los individuos con “calidades” estéticas más interesantes o según cualquier otro criterio funcional y/o estético de supervivencia. Los observadores intervienen en el diseño mediante una interfaz interactiva sin tener conocimiento de los algoritmos computacionales necesarios para realizar el proceso. Debido a esto último y aunque la usabilidad de cualquier herramienta de Diseño Interactivo Evolutivo es completa por defecto, la interfaz debe diseñarse de la forma más intuitiva, sencilla y directa posible.

Este nuevo concepto contrasta con el papel del usuario antes de la aparición de este tipo de Diseño Evolutivo ya que se limitaba a la definición de una representación adecuada del problema, la adopción de un criterio de evaluación de la aptitud de los individuos y al diseño de operadores para la optimización de los mismos sin participar en modo alguno en el proceso de evolución.

En general, el Diseño Interactivo Evolutivo es capaz de generar soluciones diversas e interesantes potencialmente competitivas para una amplia gama de problemas. Además, la interacción humana puede proporcionar información al usuario para una mayor comprensión y definición del problema en entornos mal definidos e inciertos y al mismo tiempo identifica la mejor dirección para la mejora del carácter innovador y creativo del diseño en investigaciones futuras [Parmee, 01].

Sin embargo, no todo son ventajas, el modelo puede no ser adecuado para requerimientos cambiantes y el usuario en este caso debe agregar nuevas restricciones para generar soluciones satisfactorias [Inoue, 99]. Por otro lado, la variabilidad de las preferencias personales respecto a un mismo usuario es todo un handicap. La calidad que un usuario asigna a un diseño que sobrevive varias generaciones no se mantiene invariable pues es relativa a las correspondientes a los diseños de su población. Además, la calidad de un mismo diseño depende de las preferencias del usuario que lo evalúe aún cuando el criterio aplicado sea el mismo generando un gran abanico de posibles soluciones.

Otro problema que es necesario tener en cuenta es el tamaño de la población limitada por el cansancio del usuario, aún cuando la interfaz presente una elevada usabilidad y por la limitación intrínseca humana a la hora de recordar imágenes o sonidos desplegados de manera secuencial. Por este motivo el diseño de la interfaz es tan importante.

Otro problema añadido es que el éxito del proceso evolutivo de diseño se basa en el mantenimiento de la diversidad genética para el que es necesario trabajar con poblaciones relativamente grandes. Sin embargo, esto no siempre es posible, de hecho en problemas en donde la evaluación de los individuos depende del ser humano (Diseño Interactivo Evolutivo) y en otros que implican un coste computacional alto y que requieran de respuestas rápidas e interactivas, es necesario trabajar con poblaciones pequeñas. Así, a la complejidad que conlleva la asignación subjetiva por parte del usuario de valores numéricos a los diseños generados por el sistema se suman los problemas que surgen al utilizar algoritmos con micropoblaciones dado que puede ocurrir que no exista diversidad genética y el algoritmo obtenga resultados dispares en pocas generaciones. En este sentido existen soluciones que apuestan por el uso de una función de evaluación mixta donde se realiza una selección previa de los individuos

antes de que el usuario haga la suya propia. En estos casos a la aptitud de las soluciones parciales evaluadas por el usuario se suma la obtenida por algún sistema de evaluación analítica, por ejemplo, un simulador. Esta función de evaluación combinada se compone de dos aptitudes: la aptitud implícita, también llamado fitness automático [Machado, 02] y un valor de aptitud explícita definido por el usuario.

En algunos casos el objetivo de la evolución de los diseños, considerados como seres de vida artificial, es generar un aumento de su complejidad visual. Sin embargo lo que se pretende, en general, es que éste sea especificado directamente por el usuario interactuando con el sistema. En esta interactividad se visualiza en pantalla una población de propuestas y mediante la interfaz el observador puede decidir, según sus criterios estéticos, cuál de los diseños tendrá más o menos probabilidad de supervivencia. El proceso de evolución genera descendientes como variaciones y/o combinaciones de los diseños de la generación anterior. La aplicación de estos operadores produce nuevas propuestas estéticas.

La aparición de otros trabajos de investigación en esta misma línea dio origen a “Media art”, corriente relacionada con la creación artística que hace uso de tecnologías digitales interactivas (hipermedia, sistemas de realidad virtual, inteligencia y vida artificiales y sistemas telemáticos) para establecer un diálogo entre el público y el diseño. Aún siendo menos comunes existen también trabajos de investigación en donde se aplica el concepto de Diseño Interactivo Evolutivo a la ingeniería. H. Takagi [Takagi, 01] documenta una excelente recopilación de aplicaciones sobre Diseño Interactivo Evolutivo clasificando las aplicaciones en estos dos grandes grupos.

En cuanto al arte existen trabajos de investigación que tratan de artes gráficas y animación [Sims, 91], evolución de reglas gramaticales [McCormack, 96], recuperación de imágenes [Kato, 01], creación de imágenes 2D (sistema SIBART desarrollado por T. Unemi y K. Simas) [Unemi, 00], estructuras 3-D (software MUTATOR de W. Latham y S. Todd) [Todd, 92], generación de curvas y sonidos en un entorno 3-D [Das, 94], diseño de moda [Kim, 00], generación de hojas de estilo para sitios web [Monmarche, 99], interpretación de solos de Jazz [Biles, 94], entre otras.

Respecto a las aplicaciones en Ingeniería destacan sistemas de diseño de automóviles [Graf, 95], ingeniería de alimentos [Herdy, 97], recuperación de bases de datos [Shiraki, 96], conversión de voz [Sato, 02], aplicaciones de ayuda a la escucha [Takagi, 99], sistema de recuperación de imágenes [Cho, 02] y diseño de curvaturas de arcos arquitectónicos [Parmee, 93].

El actual desarrollo del diseño interactivo está estrechamente vinculado con la creciente tendencia a potenciar la participación múltiple humana en el proceso de diseño de un determinado proceso, sistema o producto adoptando cierto carácter abierto en donde el usuario final desempeña con su acción un papel fundamental. Esta participación de la percepción humana se fundamenta en la selección según el criterio de un conjunto de observadores, la interfaz entre los mismos y el sistema computacional de proceso y la evolución de los diseños anteriores hacia otros nuevos con mayor probabilidad de éxito.

8.3 Computación evolutiva interactiva

Las técnicas de Diseño Interactivo Evolutivo han generado numerosas aplicaciones entre las que destaca el diseño artístico y funcional de objetos bidimensionales y tridimensionales que sirven de estímulo para la creatividad de artistas e ingenieros.

La computación evolutiva aplicada al diseño de morfologías tiene dos claros precedentes: el diseño de objetos bidimensionales de Dawkins y el diseño de objetos tridimensionales que aplicó Peter Bentley. Dawkins en 1986 presentó en su obra *Biomorphs* donde el usuario decidía de forma interactiva que dibujos se asemejaban a un insecto real [Dawkins, 86]. Por otro lado, Peter Bentley presentó en “*Evolutionary designs by computers*” [Bentley, 99] su aplicación *coffee tables* donde sienta las bases teóricas y prácticas de la aplicación de técnicas evolutivas al diseño creativo y funcional de objetos tridimensionales.

A continuación se enumeran proyectos relativos al Diseño Morfológico Interactivo Evolutivo en donde el criterio de evaluación y selección es meramente artístico y personal.

- Optimización de estructuras aeronáuticas (cascos de embarcaciones) mediante algoritmos genéticos desarrollado en el Centro de Diseño e Ingeniería Computacional de la Universidad de Southampton.
- Diseño de esculturas mediante la participación subjetiva de los usuarios desarrollado por Duncan Rowland de la Universidad de Michigan en 2000.[Rowland, 00].
- Tratamiento digital de imágenes [Ngo, 93], [Sims, 94].
- Creación de composiciones musicales [Moore, 94].
- Diseño automático de figuras [Vico, 99].
- Generación de diseños artísticos [Santos, 00], [Unemi, 00].
- Generación de gestos en figuras tridimensionales orientado a mundos virtuales [Berlanga, 00].

Aunque la mayoría de las aplicaciones generadas con estas técnicas están inicialmente concebidas para dar respuesta a determinadas soluciones lo óptimo sería diseñar un sistema que tratase de generar diseños tridimensionales de propósito general en donde los usuarios determinasen que diseños son los más aptos según un criterio basado en la creatividad, funcionalidad o gusto personal y sin necesidad de conocimientos complejos acerca del proceso de síntesis.

8.4 Metodología para el Diseño Automático de Morfologías 3D

8.4.1 Objetivos

El objetivo principal de esta implementación es la integración en el módulo de simulación/interacción del entorno de diseño expuesto en esta tesis de un sistema CAD clásico y una sencilla interfaz de visualización para la síntesis de morfologías tridimensionales mediante evaluaciones múltiples y subjetivas de un conjunto de usuarios.

En el ejemplo que se propone en este capítulo, el sistema generará modelos tridimensionales definidos como instancias a un modelo matemático parametrizado. El usuario accederá a cada una de las morfologías sintetizadas por el sistema morfológico-evolutivo-interactivo y las ponderará a través de una interfaz diseñada al uso. El algoritmo genético se encargará entonces de hacer evolucionar el modelo según la calidad asignada hasta obtener un producto que se ajuste a las especificaciones definidas por el conjunto de los usuarios. Para realizar esto, se integrará al proceso de diseño un sistema CAD tridimensional interactivo.

8.4.2 Implementación.

La ejecución del entorno de diseño corre a cargo de un ordenador maestro y un cluster de 50 máquinas que harán el papel de esclavos. El maestro coordinará el proceso evolutivo de optimización. Cada individuo es un sólido tridimensional diseñado en AutoCAD automáticamente por cada esclavo y es codificado como una serie de valores reales que definen puntos en el espacio. En la primera generación el servidor creará cromosomas que corresponderán a morfologías tridimensionales aleatorias que se visualizarán y evaluarán en los clientes a través de una interfaz programada en MATLAB. De este modo, en el módulo de simulación/interacción del entorno de diseño intercolaboran por lo tanto, AutoCAD como sistema de CAD clásico y una interfaz de visualización programada.

Cada una de las poblaciones está formada por 240 individuos y cada uno de ellos está codificado, a su vez, por un cromosoma de 20 genes que definen puntos tridimensionales. El sólido está representado por una superficie que pasa por todos y cada uno de esos puntos.

Para la incorporación de AutoCAD al entorno es necesario reconfigurar su funcionamiento para que en lugar de servir como herramienta de auxilio al diseñador se transforme en una herramienta de ayuda al diseño. De este modo una vez deshabilitada su interfaz, éste necesita del auxilio de un lenguaje de programación para que lo haga arrancar, leer, traducir, modelizar y exportar en formato gráfico el cromosoma correspondiente a cada una de las morfologías. Además, una vez realizado todo esto, AutoCAD se debe cerrar automáticamente.

Para este propósito se ha programado una macro en VBA (Visual Basic Automation, lenguaje de programación de AutoCAD) para que una vez que cada cliente reciba el cromosoma que le llega desde el servidor a través de una función de programación distribuida, éste abra AutoCAD, procese los datos, cree la forma en 3D indicada y genere una imagen en formato .jpg del sólido que será incorporada a la interfaz de usuario. Una vez que cada cliente procese la información tridimensional de 8 individuos el usuario accederá a la interfaz y asignará puntuaciones de 0 a 10 a cada uno de los individuos visualizados como formas 3D.

En la siguiente figura se muestra el esquema del Entorno Evolutivo de Diseño Automático desarrollado.

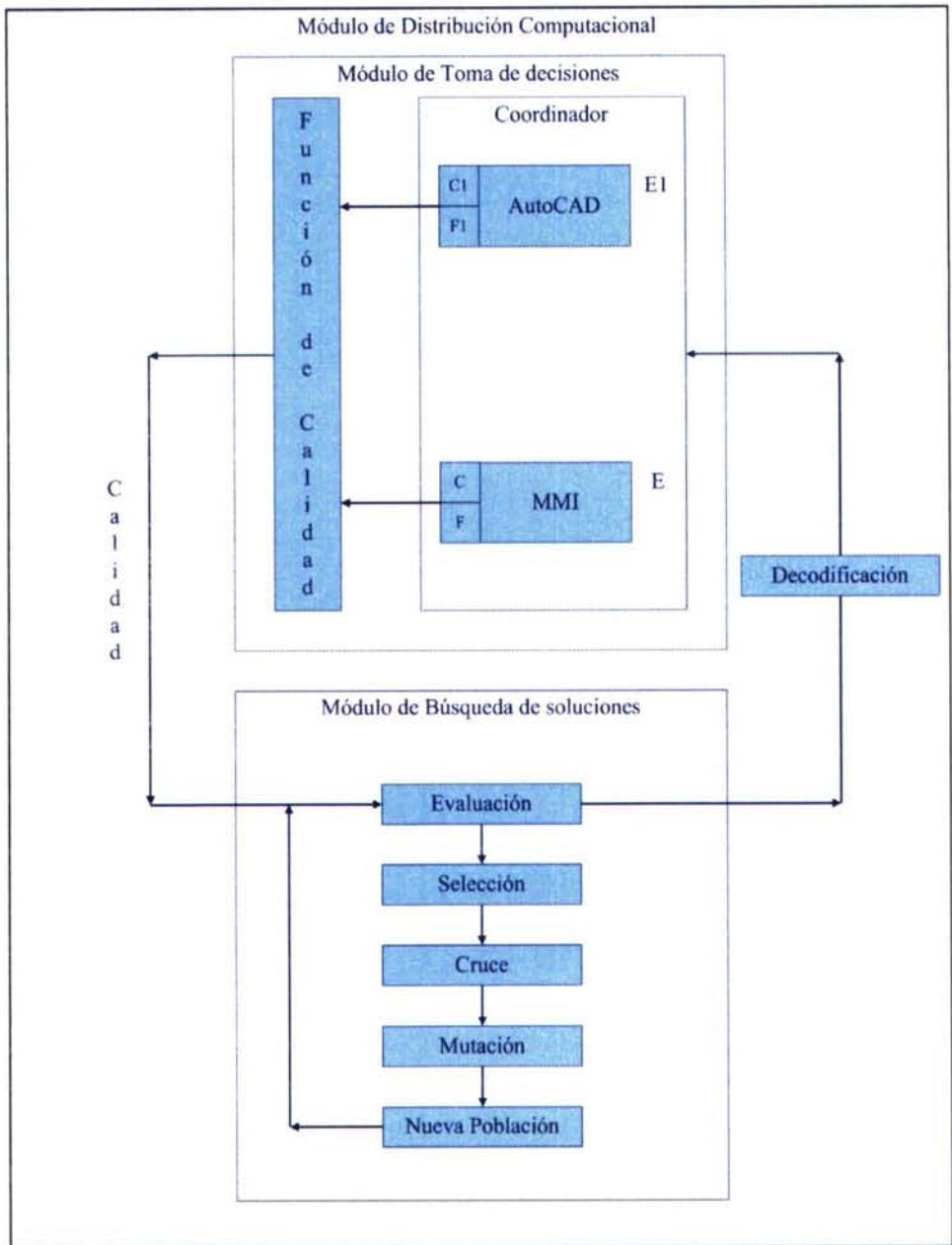


Figura 8.1: Esquema del Entorno Evolutivo de Diseño Automático implementando

En este caso particular el módulo de Toma de decisiones está constituido por un software CAD clásico (AutoCAD) y una interfaz programada en MATLAB necesaria para la evaluación subjetiva de los diseños desarrollados.

En la figura se muestra la interfaz que se ejecuta en los ordenadores responsables de la interacción con los usuarios. En cada interfaz se presentan 8 morfologías en forma de botella en 2 grupos de 4 y el usuario le otorga una calidad a cada una de ellas mediante un combobox.

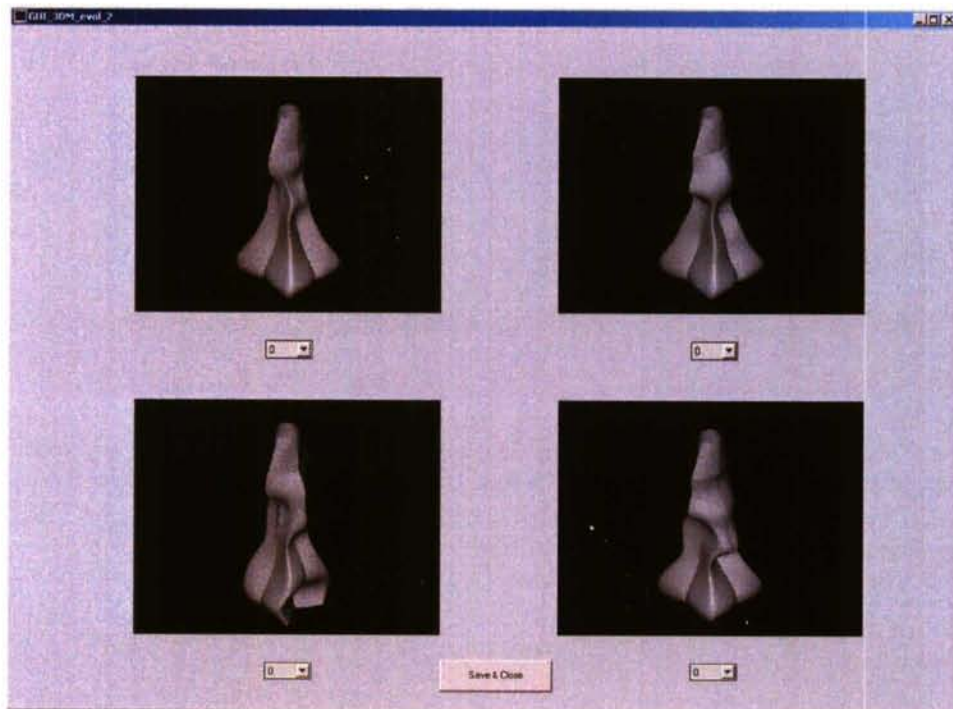


Figura 8.2: Interfaz de usuario

Una vez que el maestro recibe un mensaje que le indica que los usuarios han evaluado todos los individuos que forman una población, el maestro procesa las calidades de todos ellos y mediante operadores genéticos (selección, cruce y mutación) crea una nueva población. La probabilidad de mutación seleccionada para este caso concreto es de un 10%, siempre se produce cruce y el tipo de selección es por torneo de ventana 2. La mutación se define como la modificación de cualquiera de los genes que conforman un cromosoma y que genera una nueva morfología. Una vez que la calidad del mejor individuo de la población se asemeja a la media de las calidades de toda la población, se considera que la población ha convergido.

El experimento que se llevó a cabo consistió en dos fases diferenciadas una libre y otra guiada. Con estos dos experimentos se pretendió demostrar cómo sin especificación alguna los usuarios valoran de forma distinta cada una de las morfologías en función de ciertos criterios personales pero cuando se indica que la evaluación de los diseños debe seguir ciertos criterios, el entorno hace converger la población hacia morfologías con características similares que encajan con la descripción de la morfología deseada. A continuación se describen ambos casos.

- Caso 1

En este caso se probó el sistema ante un grupo de usuarios de 10 personas sin haber indicado un criterio de evaluación previo. La respuesta del sistema fue clara, cada uno de los usuarios tenía su criterio propio y las geometrías tridimensionales fueron tomando formas de todo tipo.

En la siguiente figura se muestra la generación inicial y el aspecto de las geometrías en la generación número 15.

Generación 1

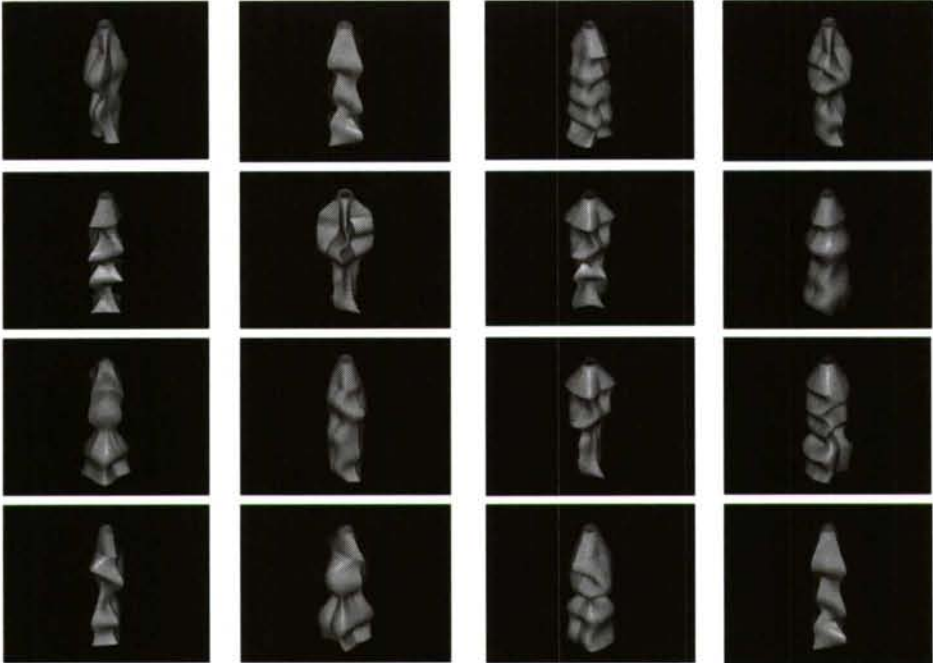


Figura 8.3: Muestra de individuos de la generación inicial

Generación 15

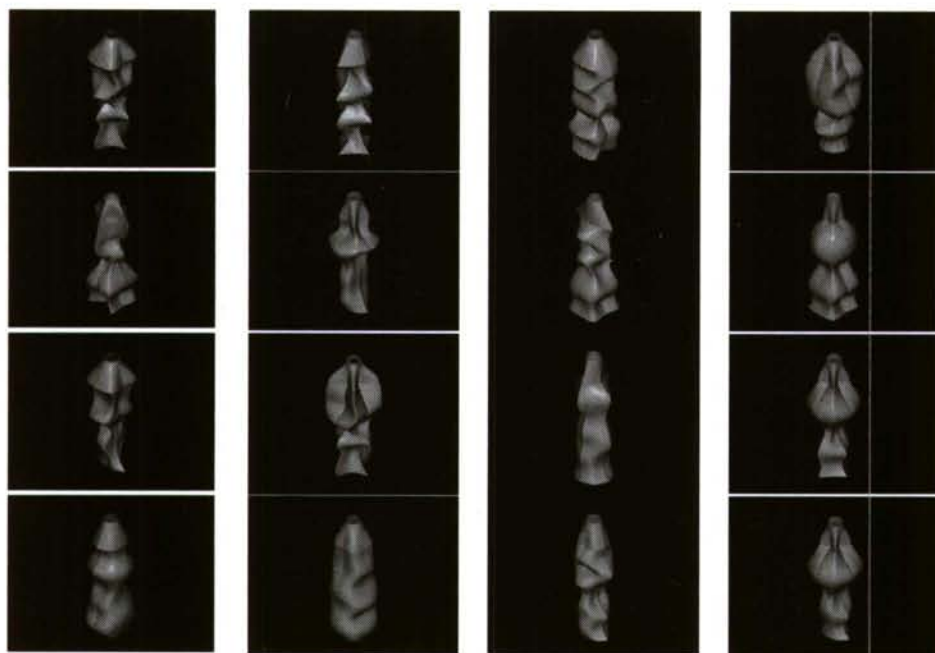


Figura 8.4: Muestra de individuos de la última generación

A la vista de los resultados se observó cómo no existía relación alguna entre las morfologías correspondientes a la última generación pues al no existir un criterio común de evaluación anteriormente pactado por los usuarios cada uno de ellos las calificó en función de un criterio personal puramente subjetivo.

- Caso 2

En el segundo experimento se definió previamente un determinado criterio de evolución que consistió en que la calificación de las geometrías generadas fuese función de su parecido con una forma piramidal genérica. Este criterio se determinó por consenso entre los usuarios de la interfaz.

En este caso se demostró que al cabo de 15 generaciones, la población de botellas se estabilizó y la mayoría de ellas adoptaban la forma deseada con ligeras modificaciones entre ellas.

Generación 1

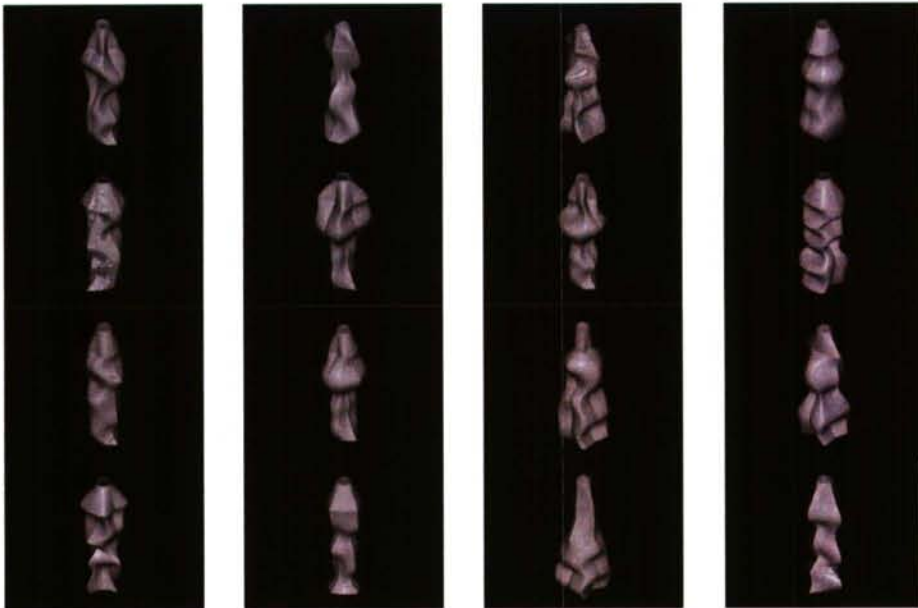


Figura 8.5: Muestra de individuos de la generación inicial

Generación 15

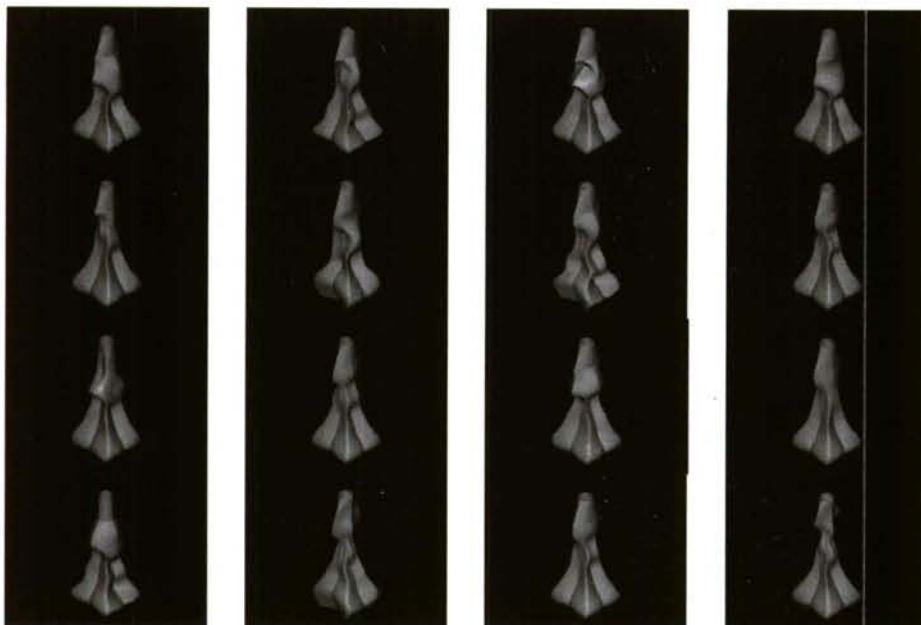


Figura 8.6: Muestra de individuos de la última generación

A la vista de los resultados se demostró cómo es posible dotar de interactividad al módulo de toma de decisiones del entorno de diseño automático para posibilitar la evaluación subjetiva de las morfologías candidatas hasta converger en una solución que concuerde con las especificaciones planteadas inicialmente (forma cónica). De este modo es posible obtener las preferencias de un conjunto de usuarios respecto a la morfología de cualquier producto para garantizar su éxito una vez salga al mercado.

Si se combina este tipo de evaluación subjetiva con otra proveniente de la evaluación técnica de las posibles soluciones es posible obtener productos que, dentro de las especificaciones técnicas y normativas existentes, presenten la morfología más ampliamente aceptada por sus usuarios finales. Al mismo tiempo es posible que además de éstos últimos intervengan en el entorno expertos que aporten su opinión técnica sobre las soluciones parciales que son diseñadas por el entorno. En definitiva el módulo de toma de decisiones puede estar formado por evaluadores subjetivos constituidos por usuarios finales o expertos que evalúen el diseño a partir de un simulador y evaluadores objetivos constituidos por simuladores de cualquier tipo (comercial o programado) y ámbito.

En relación al caso que nos ocupa, por ejemplo, es posible que la morfología correspondiese a la de un termo cuyas características mecánicas fuesen evaluadas por un simulador comercial, las térmicas evaluadas por un conjunto de expertos que evaluaran los resultados (gráficas, tablas) de un programa expresamente programado para tal fin (ad hoc) y las consistentes en la apariencia externa decididas por un conjunto de humanos que representen estadísticamente el perfil de los usuarios finales hacia los que va dirigido el producto.

8.4.3 Conclusiones del diseño morfológico

En resumen, se comprobó que es posible diseñar interactivamente morfologías tridimensionales mediante la incorporación al entorno de diseño automático de un nuevo módulo para permitir la evaluación humana de formas generadas por un sistema CAD clásico. Con este nuevo concepto se vio que era posible la combinación de criterios técnicos y subjetivos para la evaluación de posibles soluciones definiendo así un nuevo sistema de diseño sencillo y económico que se adapte al máximo a los gustos y necesidades de determinado tipo de usuarios y que posibilita el estudio de los aspectos sociológicos y subjetivos que intervienen en la aceptación de un producto.

Este nuevo concepto posibilita que el entorno pase a ser definido como un sistema sencillo y económico de “ayuda al diseño” en contraste con los sistemas de CAD clásico de “ayuda al diseñador”. De este modo es posible diseñar productos que se adapten en gran medida a los gustos y necesidades de un perfil determinado de usuarios cumpliendo al mismo tiempo las especificaciones funcionales y las normativas técnicas.

Conclusiones

9 Conclusiones

En este trabajo se parte de una revisión del desarrollo del proceso de diseño a lo largo de la historia y de las diferentes aproximaciones y escuelas que proponen metodologías para llevarlo a cabo de una forma estructurada y repetible. La mayor parte de la investigación desarrollada hasta el momento se ha centrado en la estructuración del proceso de diseño y en la optimización de ciertas etapas del mismo. En los últimos años, el trabajo desarrollado por diversos autores en la optimización del proceso de diseño ha consistido fundamentalmente en la creación de herramientas computacionales aisladas que ayudan al ingeniero en la visualización, modelado y evaluación de las posibles soluciones a un problema en función de su experiencia particular. Estas herramientas no satisfacen a la gran mayoría de los investigadores que analizan el proceso de diseño pues no tratan con sus dos fases más importantes: la búsqueda de soluciones y la toma de decisiones. Es por esto por lo que el entorno computacional de diseño que se desarrolla en esta tesis gira en torno a estas dos etapas: la etapa de búsqueda o etapa divergente donde se generan conceptualmente las soluciones y la etapa de toma de decisiones o etapa convergente en la que se realiza la evaluación y selección de las mismas.

Con respecto a la etapa de búsqueda, es evidente que si para dar solución a un problema se toma como referencia exclusivamente el conocimiento aportado por soluciones adoptadas a problemas similares se cierra el camino a otras más efectivas todavía no contempladas que podrían obtenerse a partir de una búsqueda creativa que incentivara la innovación. Sin embargo este proceso de búsqueda de soluciones innovadoras no es tan directo como el tradicional y, como consecuencia, la síntesis y evaluación de soluciones parciales requiere de un gran esfuerzo. Es en este punto donde entra en juego el objetivo principal de esta tesis: la definición de un entorno de diseño automático de sistemas y productos basado en técnicas de búsqueda avanzada que integre todas y cada una de las etapas del proceso de diseño en mayor o menor medida (definición y análisis de un problema, síntesis y análisis del diseño y especificación de la solución) y siempre teniendo como meta su aplicación práctica en la industria.

Las estrategias de búsqueda y análisis implementadas por el entorno emulan eficientemente el proceso de diseño en cuanto a exploración del espacio de posibles soluciones y evaluación de las mismas. De este modo se mejora notablemente el proceso de diseño tradicional humano minimizando los medios necesarios para su implementación.

Evidentemente, si la exploración del espacio de búsqueda pretende mejorar la propuesta tradicional ésta debe realizarse de una forma estructurada y automática. Lo primero debido a que es inviable plantearse una exploración completa o aleatoria de toda su dimensionalidad y lo segundo porque, aún siendo estructurada, necesita de la evaluación de gran número de soluciones sobre todo en caso de problemas de relativa complejidad.

La metodología de automatización propuesta replantea el problema de diseño de tal forma que es tratado como un problema de búsqueda de extremos en un espacio de soluciones, normalmente de alta dimensionalidad y, dado que se pretende proporcionar caminos para la obtención de soluciones originales, de dimensión variable.

El criterio de selección de los caminos de búsqueda necesarios para obtener un abanico de posibles soluciones es función de la evaluación de las mismas, debe tener en cuenta requisitos a veces contradictorios y ser capaz de obtener a partir de ellos una solución definitiva al problema propuesto. Este proceso no debe quedar exclusivamente en manos del diseñador debido a que sus limitaciones dan lugar a la presencia de restricciones ficticias, sobre todo cuando se trata con estructuras complejas, no lineales, multidimensionales o que requieran de la interacción de múltiples fuentes de conocimiento distintas.

Siguiendo estas premisas, el entorno de diseño desarrollado reduce la participación del diseñador humano a la formulación y análisis del problema (especificación), liberándolo de su función clásica de búsqueda y toma de decisiones (diseño). De este modo, se mejora la creatividad al eliminarse las restricciones ficticias introducidas por el diseñador humano, se hacen los problemas más manejables y se obtienen soluciones variadas de mayor calidad al trabajar directamente sobre parámetros en el dominio del problema.

La evaluación de la calidad de los diseños debe realizarse en función de dos tipos de parámetros distintos y complementarios: los parámetros técnicos, objetivos o “de ingeniería” y los puramente subjetivos. Los primeros están definidos por herramientas de simulación comerciales o bien por otras exclusivas “ad hoc”. Su elección depende de múltiples factores.

La integración del entorno de diseño con simuladores avanzados (mecánicos, hidrodinámicos, de control, etc.) posibilita que cada una de las propuestas de solución pueda ser evaluada en condiciones lo más cercanas posibles a la realidad. Este uso de aplicaciones software específicas y comerciales permiten ahorrar un tiempo considerable a la hora de programar herramientas propias. Cuando éstas no existen, su generalidad de aplicación los hacen excesivamente lentos o necesitan de un gran esfuerzo para su integración se deja abierta la posibilidad de introducir herramientas y programas desarrollados ad hoc, que pueden resultar más versátiles y fáciles de integrar necesitan de un conocimiento en profundidad del sistema a analizar.

Los parámetros de calidad subjetivos, por otro lado, requieren de la existencia de una interfaz hombre-máquina que ponga en contacto al usuario con el entorno de forma natural y transparente.

La necesidad de que el entorno sea integrable con simuladores e interfaces para asegurar la versatilidad necesaria que desarrolle una amplia gama de diseños con la mínima reconfiguración y la división del proceso de diseño en cinco etapas fundamentales da a entender que la estructura modular del entorno puede aportar ciertas ventajas. Por este motivo se ha dedicado un gran esfuerzo en definir los módulos que componen el sistema y en definir sus interfaces de manera que el entorno fuese razonablemente fácil de adaptar a nuevas propuestas.

En los casos en que la reconfiguración del entorno sea necesaria ésta consistirá, por ejemplo, en adaptar el software de evaluación y/o modificar parámetros en la metodología de búsqueda en función de las especificaciones definidas.

A pesar de que el entorno automático de diseño planteado integra estrategias de búsqueda avanzada de soluciones y herramientas de simulación cada vez más rápidas, en el caso de tratar con problemas de relativa complejidad, es necesario de una elevada

potencia de cálculo computacional. Por lo tanto el sistema debe diseñarse para que sea escalable en cuanto a los recursos computacionales que utiliza. Por esta razón hemos desarrollado una implementación contemplando su distribución computacional con distintos niveles de granularidad según cada problema en concreto.

La combinación en el entorno de diseño de una estrategia de exploración avanzada, simuladores avanzados de alta fiabilidad, una interfaz de usuario para la evaluación subjetiva de soluciones parciales y la potencia de la computación distribuida hace cambiar radicalmente el uso de las herramientas computacionales en el proceso de diseño. De este modo se pasa de un concepto inicial de herramientas de “ayuda al diseño”, que aunque cada vez más avanzadas se reducen fundamentalmente a software de análisis y visualización, a un nuevo sistema de “ayuda al diseñador” que sustituye al humano y lo relega a una función meramente especificadora y que controla el proceso de búsqueda y síntesis de posibles soluciones a partir de las especificaciones y recursos de que se disponen.

A partir de estos requisitos y después de realizar una revisión de la bibliografía relacionada con el proceso de diseño y con cada una de sus etapas, se determinaron las herramientas y algoritmos computacionales existentes relacionados con el mismo y que cumplen con las especificaciones planteadas para integrarlas de forma eficiente y conformar un entorno de diseño que se concretó en siguientes bloques fundamentales:

- Módulo de Búsqueda de soluciones.
- Módulo de Toma de decisiones (evaluación con o sin interacción humana).
- Módulo de Distribución computacional.

Se estudiaron las posibles implementaciones para cada uno de estos módulos y se optó por la implementación de un módulo de búsqueda de soluciones basado en técnicas evolutivas. En el trabajo presentado en esta tesis se ha recurrido por generalidad y simplicidad a un Algoritmo Genético Simple con pequeñas variaciones en sus operadores en función de la codificación definida en cada problema. La implementación modular asegura que la sustitución de operadores genéticos resulte sencilla e inmediata.

El módulo de toma de decisiones necesita un decodificador que transforme el genotipo en fenotipo para que, a partir de este último, una serie de simuladores (comerciales o no) e interfaces hombre-máquina evalúen técnica o interactivamente cada uno de los diseños parciales generados por el entorno. Una vez calculadas las aptitudes de cada uno de los simuladores, se comparan sus valores en función de los criterios existentes. La selección y ponderación relativa de los mismos son resultado de la etapa de especificación y se expresan en términos del dominio del problema (rendimiento económico, capacidad de producción, etc.) y no del dominio de implementación (consumos eléctricos, parámetros de control, etc.) como ocurre en el diseño tradicional.

La cuantificación de los criterios en el dominio del problema (seguridad, facilidad de manejo u operabilidad, aceptación del público, etc.) y su ponderación cuando existen conflictos suele resultar difícil. De entre ellos, el criterio más complicado es la calidad estética al ser función de la subjetividad del observador y requerir de gran variedad de respuestas en muchos casos difícilmente traducibles a criterios en el dominio de implementación.

En último término, a partir de la aptitud global de cada diseño calculada mediante una función de calidad, el módulo de toma de decisiones selecciona el camino a seguir o la solución adoptada.

Respecto a los simuladores implementados los más utilizados en esta tesis han sido Matlab, Simulink, DADS, Extend y AutoCAD.

Los dos primeros, ambos de Mathworks, y su toolbox de Control se han incorporado al entorno de diseño debido a su versatilidad y amplia utilización en ingeniería. Ambos se han usado en colaboración con DADS (simulador dinámico de CADSI) para la síntesis de controladores de maniobra y estabilización de robots submarinos. DADS es un simulador dinámico comercial de altas prestaciones que permite exportar como una caja negra las ecuaciones de movimiento de cualquier sistema mecánico e integrarlas en Simulink. A pesar de esta ventaja, ha sido necesario programar la automatización de este proceso de exportación y su ejecución automática y paralela. Extend, por otro lado, es un simulador de procesos discretos fácilmente programable para su automatización en Visual Basic Automation (VBA) pero que no está preparado para ejecutarse de forma paralela ni integrarse con herramientas de control avanzado. AutoCAD presenta las mismas características que Extend. En cuanto a que es permite su programación externa desde VBA pero no está preparado para su ejecución paralela.

En definitiva, a pesar de que los simuladores cuidan cada vez más su integración con otros sistemas y muchos de ellos son programables con VBA (Extend y AutoCAD), la mayoría no pueden ejecutarse de forma automática y paralela pues están concebidos para ser utilizados por un usuario y siempre a través de su interfaz gráfica. Por lo tanto, la incorporación de los simuladores al módulo de toma de decisiones del entorno de diseño es diferente en cada caso y supone a menudo un gran esfuerzo para la encapsulación de éstos módulos, si bien es cierto que una vez adecuadamente encapsulados son reutilizables para múltiples tipos problemas.

La evaluación interactiva consiste en una interfaz hombre-máquina que posibilita la evaluación múltiple y remota de las soluciones parciales que genera el entorno de diseño por un equipo de expertos o usuarios finales. Como consecuencia, es posible conformar una marco función de calidad constituida por ingenieros expertos, simuladores para el cálculo de propiedades técnicas y un conjunto de usuarios que representa el perfil del público al que va dirigido el producto. En este tipo de ingeniería concurrente la función de calidad trataría desde evaluaciones puramente estéticas hasta las consistentes en especificaciones y normativas técnicas.

Finalmente, el bloque de distribución de computación, encargado de dotar de escalabilidad al sistema aprovechando las características de los algoritmos evolutivos para su distribución y paralelización, se implementó utilizando un entorno de paso de mensajes estandarizado: MPI.

Para demostrar la versatilidad del entorno modular expuesto se han implementado procesos de diseño de sistemas reales en tres ámbitos completamente distintos y que han sido seleccionados por pertenecer a proyectos de investigación en plantas reales. Estos sistemas complejos se caracterizan por el acoplamiento no lineal existente entre los parámetros que definen la calidad de sus soluciones parciales en el dominio del problema, apenas relacionados con los correspondientes a sus subsistemas, en el dominio de la solución.

Se han diseñado dos tipos distintos de controladores neuronales para la estabilización y maniobra de vehículos sumergibles cuya direccionabilidad se basa en thrusters y superficies de control respectivamente. La dinámica del medio y el sistema y la dificultad de modelado de las interacciones existentes dan lugar a la necesidad de sintetizar controladores no lineales de gran complejidad. El primer tipo de controladores permite la estabilización en posición de un vehículo sumergible autónomo con brazo manipulador ante la presencia de cualquier fuerza ejercida sobre el sistema, incluyendo las perturbaciones externas y las reacciones provocadas por el movimiento del brazo. Los controladores no lineales generados por el entorno de diseño actúan sobre los propulsores del vehículo en función de las señales que reciben desde un sistema de detección de tal modo que la posición del submarino permanece inalterable.

Se trató el diseño preliminar de un submarino autónomo que operaba con y sin brazo excavador, introduciendo así el problema dinámico de dos cuerpos acoplados. Se estudiaron los controladores resultantes comprobando las respuestas del sistema ante distintos tipos de perturbaciones para una maniobra de hovering con y sin manipulador y se demostró que cumplían con las especificaciones de diseño. Los resultados obtenidos permitieron en un desarrollo posterior posicionar adecuadamente los thrusters y establecer el diseño del sistema de control de un AUV real para exploración.

El segundo tipo de controladores se diseñó para la maniobra de un catamarán sumergible de 20 toneladas para uso turístico. El controlador solución debía compensar las fuerzas perturbadoras externas que actuasen sobre el vehículo y adaptarse a las trayectorias y consignas especificadas dentro de los parámetros de seguridad y confort adecuados mediante su actuación sobre un sistema de propulsión (thruster) y guiado (timones de profundidad y dirección).

En este caso es de destacar que al mismo tiempo que la síntesis de cada controlador entrañaba cierta dificultad debido a las características del medio submarino y la dinámica de este tipo de vehículos (la respuesta cambia bruscamente en función de la velocidad, orientación y tipo de maniobra), fue necesario tener presente que su dinámica no lineal dio lugar a que sus controladores presentasen respuestas acopladas entre sí.

Para la evaluación de los distintos diseños obtenidos se utilizó un módulo de simulación realizado en Simulink por otro miembro del grupo de investigación que permitía evaluar la respuesta del conjunto de controladores. La ejecución distribuida del entorno se realizó utilizando un cluster de 50 ordenadores.

Los controladores generados por el entorno además de cumplir con todas las especificaciones definidas permitían evitar problemas de balanceo mediante una adecuada compensación de las velocidades y ángulos de los timones de profundidad y dirección. Este resultado es importante desde el punto de vista que el único medio que presentaba el catamarán inicialmente para realizar esta compensación consistía en la fuerza restauradora que generaba la posición relativa de su centro de masas respecto al de flotación, fuerza cuya actuación es muy lenta.

Del mismo modo se estudió mediante la metodología que se expone en esta tesis la disposición y control de maquinaria en líneas de producción industriales haciendo intervenir parámetros de carácter general en el criterio de búsqueda de soluciones, tales como la productividad, que presentan una relación extremadamente indirecta con los correspondientes a sus subsistemas.

Respecto a la optimización de la disposición de maquinaria se definió una línea de ensamblaje de dos subproductos complementarios y cadencias distintas con la intención de sintetizar la disposición óptima de *buffers* de modo que con la menor energía posible no existiesen atascos ni pérdidas. En este caso el entorno obtuvo una disposición que permitía que la línea alcanzase su máxima productividad con el mínimo coste.

Como segundo ejemplo, y esta vez con la limitación de que la planta ya existía, se trató de obtener una estrategia de control que evitase la aparición de atascos en una línea de barnizado real. Para ello se aplicó el entorno de diseño con el objetivo de sintetizar un controlador secuencial que gestionase los transportes de entrada y salida de la línea y así optimizar su productividad minimizando al mismo tiempo el gasto energético del proceso. Los circuitos lógicos secuenciales de cada controlador se sintetizaron a partir de puertas lógicas nand y circuitos flip-flop y se codificaron en forma de árbol binario. La actuación del controlador se realiza sobre la velocidad de los transportes de la línea en función de las señales que aporta el sistema de detección de la planta minimizando las esperas y/o atascos en la misma.

En el estudio de este caso, y a modo de comparación, se realizó el análisis y el diseño de dos formas distintas. En primer lugar se siguió el método tradicional de estudio de la problemática de la planta para buscar a continuación una solución próxima al lenguaje del controlador a implementar (velocidades de tapete, rodillo, etc....) para después implementar el controlador. Por otra parte se generó el diseño automáticamente utilizando el entorno desarrollado y a través de parámetros provenientes del dominio del problema y plasmados en el documento de especificación (productividad, atascos, etc...). En este caso, la evaluación de los posibles diseños se realizó utilizando un modelo de la planta implementado en EXTEND apropiadamente encapsulado a través de un interfaz realizado en VBA.

Los resultados de este ejemplo demostraron que el sistema de diseño automático alcanzaba consistentemente soluciones tan buenas o incluso mejores que las obtenidas mediante el análisis tradicional. De hecho, los controladores resultantes (circuitos electrónicos digitales) hacían mejor uso de las señales de los sensores para adaptarse al proceso y evitar los atascos en la línea.

Con el objeto de estudiar la forma de introducir parámetros subjetivos en entornos de este tipo se abordó el diseño interactivo de morfologías mediante la incorporación de un nuevo módulo que permitió la evaluación humana de formas tridimensionales generadas por un sistema CAD clásico. Con este nuevo concepto y tomando como punto de partida el concepto de ingeniería concurrente se vio que era posible la combinación de criterios técnicos y subjetivos para la evaluación de posibles soluciones definiendo así un nuevo sistema de diseño sencillo y económico que se adapte al máximo a los gustos y necesidades de determinado tipo de usuarios y que posibilite el estudio de los aspectos sociológicos que intervienen en la aceptación de un producto. De este modo el diseñador puede participar de manera no exclusiva en la evaluación de determinadas soluciones mediante la aportación de sus conocimientos técnicos al sistema o para hacerse usuario del mismo.

En síntesis, la automatización del proceso de diseño de sistemas y productos que se propone se basa en la transformación del problema de diseño en un problema de búsqueda de soluciones mediante la aplicación de herramientas computacionales de búsqueda y evaluación. Las primeras están representadas por técnicas evolutivas debido a su flexibilidad y versatilidad y las segundas por simuladores avanzados de alta

fiabilidad que se ejecutan mediante programación distribuida para aumentar la potencia de cálculo y dotar al sistema de escalabilidad. Estos simuladores integrados en el entorno pueden realizar la evaluación técnica y automática de cada una de las soluciones parciales sintetizadas o visualizarlas y permitir la opinión subjetiva de expertos o usuarios finales mediante una interfaz hombre-máquina. Esta incorporación de parámetros subjetivos confiere al entorno de cierto carácter concurrente que podría implementarse en el trabajo futuro ayudado por plataformas como Internet.

El trabajo expuesto en esta memoria no pretende ser la solución definitiva al problema del diseño automático si no una indicación conceptual para la especificación de un sistema que permita generar soluciones más o menos originales, en función de la aplicación en el método de búsqueda de soluciones previas y siempre susceptibles de ser generadas mediante la interacción con el humano que interpreta el papel de usuario final o diseñador. De esta manera se integran en el proceso aspectos tradicionales de diseño y otros nuevos relacionados con la aceptación del producto y la ingeniería concurrente.

Una vez planteadas las principales líneas de actuación del entorno definido es cuando se puede comenzar a estudiar en concreto cada uno de sus módulos y llevarlos al nivel de operatividad y fiabilidad que permitan su introducción en el campo de la ingeniería y así pasar del concepto de diseño asistido por computador (CAD) tradicional, al diseño mediado por computador (CMD), donde el ordenador es el que realiza el diseño y el humano un simple evaluador más dentro de un conjunto de herramientas de asistencia.

Referencias

10 Referencias

- Abel, J.M., "Cable Interactions in a Depth Controlled Submersible", *Journal of Hydronautics*, vol. 6 no. 2, pp. 83-89, 1972.
- Accreditation Board for Engineering and Technology, *Annual Report*, October 1986.
- Adamidis, P., "Parallel Evolutionary Algorithms: A Review", *HERCMA '98*, Atenas, 1998.
- Adeli, H., Cheng, N., "Concurrent Genetic Algorithms for Optimization of Large Structures", *ASCE Journal of Aerospace Engineering* v7:3, pp. 276-296, 1994.
- Alabau, M., Idoumghar, L., Schott, R., "New Hybrid Genetic Algorithms for the Frequency Assignment Problem", *Proceedings of the 13th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'01)*, 2001.
- Alba, E., Troya, J.M., "Analyzing synchronous and asynchronous parallel distributed genetic algorithms", *Journal "Future Generation Computer Systems"*, volume 17, pp. 451-465, 2001.
- Alexander, C., "Notas sobre la síntesis de la forma", Cambridge Unieversity Press, 1964.
- Alexander, C., "Design Methods Group Newsletter", Vol. 5, No. 3, 1971.
- Alger, J.R.M. , Hays, C.V., "Creative Synthesis in Design", Prentice-Hall, Englewood Cliffs, NJ, 1964.
- Antonelli, G., Chiaverini, S., "Task-priority redundancy resolution for underwater vehicle-manipulator systems", *Proceedings of IEEE ICRA*, Leuven, Belgium, pp. 768-773, 1998.
- Archer, B., "Método sistemático para diseñadores", *Design*, vol. 64, 1963.
- Ariza, C., "Prokaryotic Groove: Rhythmic Cycles as Real-Value Encoded Genetic Algorithms", *Proceedings of the International Computer Music Conference*, 2002.
- Asimov, M., "Introducción to design". Games B. Reswick, N.Yersey, 1962
- Axelrod, R.M., "The Evolution of Strategies in the Iterated Prisoner's Dilemma", *Genetic Algorithms and Simulated Annealing*, ch. 3, pp 32-41, 1987.
- Axelrod, R.M., Wu, J., "How to Cope with Noise in the Iterated Prisoner's Dilemma", *Journal of Conflict Resolution*, 39, pp. 183-189, 1995.
- Babcock, P.S., Zinchuck, J.J., "Fault Detection Design Optimization: Application to an Autonomous Underwater Vehicle Navigation System", *Proceedings IEEE AUV 90*, Washington D.C., 1990.
- Bäck, T., "Evolutionary Algorithms in Theory and Practice", Oxford University Press, New York, 1996.

- Bagley, D., "The Behavior of Adaptive Systems Which Employ Genetic and Correlation Algorithm", *Doctoral Dissertation*, The University of Michigan, 1967.
- Baker, J.E., "Reducing bias and inefficiency in the selection algorithm", *Proceedings of the Second International Conference on Genetic Algorithms and Their Applications*, pp. 14-21, 1987.
- Baldacci, R., Hadjiconstantinou E., Maniezzo V., Mingozzi A., "A new method for solving capacitated location problems based on a set partitioning approach", *Computers & Operations Research* Vol 29, Issue 4, pp. 365-386, 2002.
- Baldasano, J.M., "Influencia de la Informática en el Proceso de Proyectar", *V Encuentro Nacional de Proyectos de Ingeniería, UPV*, pp. 20-21, 1989.
- Baron, P., Fisher, R., Mill, F., Sherlock, A., Tuson, A., "A Voxel-based Representation for the Evolutionary Shape Optimisation of a Simplified Beam: A Case-Study of a Problem-Centred Approach to Genetic Operator Design", *2nd On-line World Conference on Soft Computing in Engineering Design and Manufacturing (WSC2)*, 1997.
- Beer, R.D., Gallagher, J.C., "Evolving Dynamical Neural Networks for Adaptive Behavior", *Adaptive Behavior*, Vol. 1, N. 1, pp. 91-122, 1992.
- Benjamin, W., "La obra de arte en la era de su reproductibilidad técnica", *Discursos Interrumpidos, I. Madrid: Taurus Ediciones*, pp. 40, 1973.
- Bentley, P.J., Wakefield, J.P., "Generic Representation of Solid Geometry for Genetic Search", *Microcomputers in Civil Engineering* 11:3, Blackwell Publishers, pp. 153-161, 1996a.
- Bentley, P.J., Wakefield, J.P., "The Evolution of Solid Object Designs using Genetic Algorithms", Rayward-Smith, *Modern Heuristic Search Methods*. Ch. 12, John Wiley & Sons Inc., pp. 199-215, 1996b.
- Bentley, P.J., Wakefield, J.P., "Conceptual Evolutionary Design by Genetic Algorithms", *Engineering Design and Automation Journal* v3:2, John Wiley & Sons, Inc, pp. 119-131, 1997a.
- Bentley, P.J., Wakefield, J.P., "Finding Acceptable Solutions in the Pareto-Optimal Range using Multiobjective Genetic Algorithms", *Soft Computing in Engineering Design and Manufacturing*, Springer Verlag London Limited, Part 5, pp. 231-240, 1997b.
- Bentley, P.J., Wakefield, J.P., "Generic Evolutionary Design", *Soft Computing in Engineering Design and Manufacturing*. Springer Verlag, Part 6, pp. 289-298, 1997c.
- Bentley, P.J., "The Revolution of Evolution for Real-World Applications", *Emerging Technologies '97: Theory and Application of Evolutionary Computation*, 15th December, University College London, 1997d.
- Bentley P.J., "From Coffee Tables to Hospitals: Generic Evolutionary Design", *Evolutionary design by computers*, Morgan-Kaufman, pp. 405-423, 1999.

Bentley, P. J., "Natural Design by Computer", *Proc of the AAAI Symposium on Computational Synthesis*, Stanford University, California, 2003.

Berlanga, A., Isasi, P., Segovi, a J., "Interactive Evolutionary Computation with Small Population to Generate Gestures in Avatars", *Proceedings of GECCO 2001, Artificial Life, Adaptive Behavior and agents*, 2000.

Biles, J. A., "GenJam: A genetic algorithm for generating jazz solos," *Intl. Computer Music Conf. (ICMC'94)*, Aarhus, Denmark, pp. 131-137, 1994.

Bledsoe, W., "The use of biological concepts in the analytical study of systems", *Proceedings of the ORSA-TIMS National Meeting*, San Francisco, 1961.

Blumberg, A.F., Signell, R.P., Jentner, H.L., "Modeling Transport Processes in the Coastal Ocean," *Environmental Science in the Coastal Zone: Issues for Further Research*, National Research Council Commission on Geosciences, Environment and Resources, National Academy Press, Washington, 1994.

Boden, M.A., "The Creative Mind : Myths & Mechanisms", *Basic Books Pub.* ISBN: 0465014518, 1992.

Bonabeau, E., Theraulaz, G., Arpin, E., Sardet, E., « The Building Behaviour of Lattice Swarms", *Artificial Live IV, Proc. of the 4th Int. Workshop on the Synthesis and Simulation of Living Systems*, MIT Press, pp. 307-312, 1994..

Bono, E., "El pensamiento lateral", Barcelona, Ediciones Paidós, 1991.

Bonsiepe, G., "El diseño de la periferia", Barcelona, Ed. Gustavo Gili, 1985.

Bouchard, E. E., Kidwell, G. H., Rogan, J. E., "The application of artificial intelligence technology to aeronautical system design", *AIAA/AHS/ASEE Air-craft Design Systems and Operations Meeting*, Atlanta, AIAA, paper 88-4426, 1988.

Box, G.E.P., "Evolutionary operation: a method for increasing industrial productivity", *Applied Statistics*, 6 pp. 81-101, 1957.

Brancart, C.P., "The Evolution of the DARPA/ARPA UUV Program," *International Advanced Robotics Programme (IARP): Mobile Robots for Subsea Environments*, Monterey Bay Aquarium Research Institute, pp.12, 1994.

Brayton, R.K., Hachtel, G.D., McMullen, C.T., Sangiovanni – Vincentelli A.L., "Logic Minimization Algorithms for VLSI Synthesis", *Kluwer Academic Publishers, Dordrecht*, The Netherlands, 1984.

Brayton, R.K., Rudell R., Sangiovanni-Vincentelli A., Wang A.R., "Mis: A multiple-level logic optimization system", *IEEE Transactions on Computer-Aided Design CAD-6*, 6:1062–1081, 1987.

Bremermann, H.J., "Optimization through evolution and recombination", *Self Organizing Systems*, Washington, DC: Spartan, pp. 93-106, 1962.

Brutzman, D.P., "From virtual world to reality: designing an autonomous underwater robot", *AAAI Fall Symposium on Applications of Artificial Intelligence to Real-World Autonomous Mobile Robots*, Massachusetts, pp. 18-22, 1992.

- Brutzman, D.P., "NPS AUV Integrated Simulator, Masters Thesis", Naval Postgraduate School, Monterey California, 1992.
- Brutzman, D.P., "Beyond intelligent vacuum cleaners", *American Association for Artificial Intelligence (AAAI) Fall Symposium on Applications of Artificial Intelligence for Instantiating Real-World Agents*, North Carolina, pp. 23-25, 1993.
- Brutzman, D.P., "Video Proceedings of the Eighth International Symposium on Unmanned Untethered Submersible Technology", University of New Hampshire, Durham, New Hampshire, 1993.
- Brutzman, D.P., "Video Proceedings of the IEEE Oceanic Engineering Society 1994 Autonomous Underwater Vehicle Conference", Charles Stark Draper Laboratories, Cambridge Massachusetts, Includes two self-produced video segments: "A Virtual World for an Autonomous Underwater Vehicle" and "MBone: Multicast Backbone Audio/Video Tools for International Collaboration.", 1994.
- Burton, A. R., Vladimirova, T., "A Genetic Algorithm Utilising Neural Network Fitness Evaluation for Musical Composition", *International Conference on Genetic Algorithms and Artificial Neural Networks*, 1997.
- Cai, J. & Thierauf, G., "Structural Optimization of a Steel Transmission Tower by using Parallel Evolution Strategy", *Proceedings of Adaptive Computing in Engineering Design and Control - '96*, pp. 18-25, 1996
- Cantú-Paz, E., "Efficient and Accurate Parallel Genetic Algorithms", Kluwer Academic Publishers, 2001.
- Canudas-de-Wit, C., Diaz, E.O., Perrier, M., "Robust nonlinear control of an underwater vehicle/manipulator system with composite dynamics", *Proc. IEEE ICRA*, Leuven, Belgium, pp. 452-457, 1998.
- Coello, C.A., Zavala, R.L., Benito, G., Hernández, A., "Ant Colony System for the Design of Combinational Logic Circuits", *Evolvable Systems: From Biology to Hardware*, Edinburgh, pp. 21-30, 2000.
- Coello, C.A., "An Empirical Study of Evolutionary Techniques for Multiobjective Optimization in Engineering Design", PhD thesis, Department of Computer Science, Tulane University, New Orleans, 1996.
- Chakrabarti, A., "FuncSION: a software for synthesis of mechanical designs", *Engineering Computing Newsletter*, issue 59, 1-2, 1995.
- Chen, C., Trivedi, M.M., Bidlack, C.R., "Simulation and Animation of Sensor-Driven Robots", *IEEE Transactions on Robotics and Automation*, vol. 10 no. 5, pp. 684-704, 1994.
- Cho, S.B., Lee, J.Y., "A human-oriented image retrieval system using interactive genetic algorithm", *IEEE Transactions on Systems, Man and Cybernetics*, Part A, Volume: 32 Issue: 3, pp 452 -458, 2002.
- Choi, S.K., Takashige, G.Y., Yuh, J., "Experimental Study on an Underwater Robotic Vehicle: ODIN", *Proceedings of the IEEE Oceanic Engineering Society Conference Autonomous Underwater Vehicles (AUV) 94*, Cambridge, pp. 79-84, 1994.

- Choi, S.K., Yuh, J., Takeshige, G., "Development of the Omnidirectional Intelligent Navigator", *IEEE Robotics & Automation Mag*, 2, pp. 44-53, 1995.
- Cliff, C., Husbands, P., Meyer, J., Wilson, S.W., "From Animals to Animats 3", *Proceedings of the third international conference on simulation of adaptive behaviour*, MIT Press, 1994.
- Cliff, D.T., Husbands, P., Harvey, I., "Evolving Visually Guided Robots", *From Animals to Animats 2*, MIT Press Bradford Books, Cambridge, pp. 374-383, 1993.
- Coates, P., "Using Genetic Programming and L-Systems to explore 3D design worlds", *CAAD Futures '97*, Kluwer Academic Publishers, Munich, 1997.
- Coello, C.A., Hernández, A., Buckles, B.P., "Evolutionary Multiobjective Design of Combinational Logic Circuits", *Proceedings of the Second NASA/DoD Workshop on Evolvable Hardware*, pp 161-170, Los Alamitos, California, 2000.
- Coello, C.A., Christiansen, A.D., Hernández, A., "Automated Design of Combinational Logic Circuits Using Genetic Algorithms", *Proceedings of the International Conference on Artificial Neural Nets and Genetic Algorithms, ICANNGA '97*, pp. 335-338, Norwich, England, 1997.
- Cohen, J.E., Small, C., Mellinger, A., Gallup, J., Sachs, J., "Estimates of coastal populations", *Science* 278 (5341), pp. 1211-1212, 1997.
- Collins R.J., Jefferson D.R., "Selection in Massively Parallel Genetic Algorithms", *Proceedings of the Fourth International Conference on Genetic Algorithms*, 1991.
- Compton, M.A., "Minefield Search and Object Recognition for Autonomous Underwater Vehicles", Master's Thesis, Naval Postgraduate School, March 1992.
- Coons, S.A., "Surfaces for Computer Aided Design of Space Forms", *M.I.T. Project MAC TR-41*, 1967.
- Covington, J.H., "Implementing an Ocean Theater in NPSNET", Master's Thesis, Naval Postgraduate School, Monterey California, March 1994.
- Cristi, R., Caccia, M., Veruggio, G., Healey, A.J., "A Sonar Approach to AUV Localization", *CAMS'95*, Trondheim, Norway, pp. 291-298, 1995.
- Cristi, R., Papoulias, F.A., Healey, A.J., "Adaptive sliding mode control of autonomous underwater vehicles in the dive plane", *IEEE J. of Oceanic Engineering*, 15 (3): pp. 462-470, 1991.
- Das, S., Franguiadakis, T., Papka, M., DeFanti, T., Sandin D., "A Genetic Programming Application in Virtual Reality", *International Conference on Evolutionary Computation*, pp. 480-484, 1994.
- Dasgupta, D., McGregor, D.R., "Nonstationary Function Optimization using the Structured Genetic Algorithm", *Parallel Problem Solving from Nature 2*, Brussels, Belgium, Elsevier Science Pub., pp. 145-154, 1992.
- Davis, L., "Handbook of Genetic Algorithms", New York: Van Nostrand Reinhold, 1991.

- Dawkins, R., "The Blind Watchmaker", Longman Scientific & Technical Pub, 1986.
- Dawkins, R., "The Evolution of Evolvability. Artificial Life", *Proceedings of an Interdisciplinary Workshop on the Synthesis and Simulation of Living Systems*, Vol. VI, Los Alamos, New Mexico, Addison-Wesley Pub. Corp, pp.201-220, 1989.
- De Cos, M., "Teoría General del Proyecto. Ingeniería de Proyectos", Editorial Sintesis, Madrid, 1998.
- De Garis, H., "Brain Building with GenNets", *International Neural Networks Conference*, 1990.
- De Jong, K., "Analysis of the behavior of a class of genetic adaptive systems", Ph.D. Thesis, Department of Computer and Communications Sciences, University of Michigan, Ann Arbor, MI., 1975.
- De Jong, K.A., Spears, W.M., "Learning Concept Classification Rules Using Genetic Algorithms", *Proceedings of the Twelfth International Conference on Artificial Intelligence IJCAI-91*, 1991.
- Deb, K., "Binary and Floating Point Function Optimization using Messy Genetic Algorithms", *Illinois Genetic Algorithms Laboratory (IlliGAL)*, report no. 91004, 1991.
- Deb, K., Goldberg, D.E., "Sufficient Conditions for Deceptive and Easy Binary Functions", *Illinois Genetic Algorithms Laboratory (IlliGAL)*, report no. 92001. 13, 1992.
- Deb, K., Goldberg, D.E., "Analyzing Deception in Trap Functions", *Foundations of Genetic Algorithms 2*, Morgan Kaufmann Pub, 1993.
- Deb, K., Goldberg, D., Kargupta, H., Harik, G., "Rapid, Accurate Optimization of Difficult Problems Using Fast Messy Genetic Algorithm", *IlliGAL Report No.93004*, Illinois Genetic Algorithms Library, Department of General Engineering, University of Illinois, Urbana, 1993.
- DeBitetto, P.A., "Fuzzy Logic for Depth Control of Unmanned Undersea Vehicles", *Proc. AUV Symp.*, pp. 233-241, 1994.
- Descartes, R., "Discours de la méthode pour bien conduire la raison et chercher la verité dans les sciences", La géometrie, Ian Maire, Leyden, 1637.
- Dorato, P., "A historical review of robust control", *IEEE Control Systems Magazine*, pp. 44-47, 1987.
- Dorfles, G., "El diseño industrial y su estética", Barcelona, Ed. Labor, 1977.
- Draper, R.N., "Supercomputing 93", *IEEE Computational Science and Engineering*, vol. 1 no. 1, pp. 85-86, 1994.
- Dunn, S.E., Rae, G.J.S., "On-line Damage Detection for Autonomous Underwater Vehicles", *IEEE AUV'94*, pp. 383-392, 1992.

Dyer M.G., Flowers, M., Hodges, J., "EDISON: An Engineering Design Invention System Operating Naively", *Artificial Intelligence in Engineering Journal*, Computational Mechanics, Vol. 1, No. 1, pp. 36-44, 1986.

Dym, C., Levitt, R., "Knowledge-based Systems in Engineering", McGraw- Hill Inc., pp. 182-183, 1991.

Eby, D., Averill, R., Gelfand, B., Punch, W., Mathews, O., Goodman, E., "An Injection Island GA for Flywheel Design Optimization", *5th European Congress on Intelligent Techniques and Soft Computing EUFIT '97*, vol 1, pp. 687-691, 1997.

Eckhard, R., "Stan Ulam, John von Neumann and the Monte Carlo method", Los Alamos Science, 15, 1987.

Edwards, R. M., Cook, G. G., Khamas, S. K., Aidley, R. J., Chambers, B., "Design of a circularly polarised spiral antenna using a dual objective genetic algorithm", *Electronics Letters*, vol 34, no 7, pp 608-609, 1998.

Engelson, S.P., Bertani, N., "Ars Magna: The Abstract Robot Simulator Manual", version 1.0, *Department of Computer Science technical report*, Yale University, New Haven Connecticut, 1992.

Eshelman, L.J., Caruana, R., Schaffer, J.D., "Biases in the Crossover Landscape", *The 1989 International Conference on Genetic Algorithms*, 1989.

Feldman, J., "DTNSRDC Revised Standard Submarine Equations of Motion", *technical report DTNSRDC/SPD-0393-09*, David W. Taylor Naval Ship Research and Development Center (DTNSRDC), Bethesda Maryland, 1979.

Feng, Z., Allen, R., "Modelling of Subzero II", ISVR Technical Memorandum 880, Southampton, University of Southampton, 2001.

Fleming, P.J., Purshouse, R.C., "Genetic Algorithms in Control Systems Engineering", Technical Report No. 789, Department of Automatic Control and Systems Engineering, University of Sheffield, UK, 2001.

Fogel, L.J., "Biotechnology: Concepts and Application", Englewood Cliffs, NJ: Prentice Hall, 1963.

Fogel, D.B., "Evolutionary Computation, Toward a New Philosophy of Machine Intelligence", IEEE Press, New York, 1995.

Fogel, D.B., "Evolutionary Entertainment with Intelligent Agents", *IEEE Computer*, pp. 94-96, 2003.

Fonseca, C.M., Fleming, P.J., "An Overview of Evolutionary Algorithms in Multi-Objective Optimization", *Evolutionary Computation*, MIT Press, vol. 3, n.º 1, pp. 1-16, 1995.

Forrest, S., "Documentation for prisoner dilemma and norms programs that use the genetic algorithm", Technical report, University of Michigan, 1985.

Fossen, T.I., "Guidance and Control of Ocean Vehicles", John Wiley & Sons, New York, 1994.

- Fossen, T.I., "Underwater vehicle dynamics", *Underwater Robotic Vehicles: Design and Control*, Albuquerque: TSI, 1995.
- Foy, M.D., Benekohal, R.F., Goldberg D.E., "Signal Timing Determination Using Genetic Algorithms", *Transportation Research*. Record 1365, TRB, National Research Council, Washington, pp. 108-115, 1993.
- French, M.J., "Invention and Evolution: Design in Nature and Engineering", Cambridge University Press, 1994.
- French, M., Ramirez, A.C., "Toward A Comparative Study of Quarter-Turn Pneumatic Valve Actuators", *Journal of Engineering Manufacture*, part B, pp. 543-552, 1996.
- Friedman, G. J., "Digital simulation of an evolutionary process", *General Systems: Yearbook of the Society for General System Research*, 4:pp.171-184, 1959.
- Friedman, G. J., "Selective feedback computers for engineering synthesis and nervous system analogy", Master's thesis, University of California at Los Angeles, 1956.
- Fujii, T., Ura, T., "Development of an Autonomous Underwater Robot "Twin-Burger" for Testing Intelligent Behaviors in Realistic Environments", *Autonomous Robots 3*, pp. 285-296, 1996
- Funes, P., Pollack, J., "Computer Evolution of Buildable Objects", Brandeis University Computer Science Technical Report CS-97-191, 1997.
- Furuta, H., Maeda, K., Watanabe, W., "Aplication of Genetic Algorithm to Aesthetic Design of Bridge Structures", *Microcomputers in Civil Engineering v10:6*. Blackwell Publishers, MA, pp. 415-421, 1995.
- Garipov, V., Diakov, E., Semenkin, E., Vakhtel, S., "Adaptive Search Methods in Spacecraft Systems Optimal Design", *Proceedings of Adaptive Computing in Engineering Design and Control - '94*, pp.194-201, 1994.
- Gen, M., Cheng, R., "Genetic Algorithms and Engineering Design", John Wiley & Sons, USA, 1997.
- George, F., "Hybrid Genetic Algorithms with Immunisation to Optimise Networks of Car Dealerships", *Edinburgh Parallel Computing Centre*, 1994.
- Gero, J.S., "Computers and Creative Design", *The Global Design Studio*, National University of Singapore, pp. 11-19, 1996.
- Gertler, M., Hagen, G., "Standard Equations of Motion for Submarine Simulation", Naval Ship Research and Development Center (NSRDC) Research and Development Report 2510, Washington DC, June 1967.
- Girard, A.R., Smith, S.M., Ganesan, K., "A Convenient Form for Discrete Event Control of Autonomous Underwater Vehicles", *IARP 98*, pp. 125-130, 1998.
- Goheen, K.R., Jefferys, E.R. Broome, D.R., "Robust Self-Designing Controllers for Underwater Vehicles", *Transactions of the ASME*, vol. 109, pp. 170-178, 1987.

- Goheen, K.R., Jefferys, E.R., "Multivariable Self-tuning Autopilots for Autonomous and Remotely Operated Underwater Vehicles", *IEEE Journal Oceanic Engineering*, 15(3), pp. 144-151, 1990.
- Goheen, K., "Techniques for URV modelling", *Underwater Robotic Vehicles: Design and Control*, Albuquerque: TSI, 1995.
- Goldberg, D.E., "Genetic Algorithms in Search, Optimization, and Machine Learning". Addison-Wesley, Reading, MA, 1989.
- Goldberg, D.E., Rudnick, M., "Schema variance from Walsh-schema transform", *Complex Systems*, 5: pp. 265-278, 1991.
- Goldberg, D. E., Horn, J., Deb, K., "What Makes a Problem Hard for a Classifier System?", Illinois Genetic Algorithms Laboratory (IlligAL), report no. 92007, 1992a.
- Goldberg, D.E., Deb, K., Horn, J., "Massive multimodality, deception and genetic algorithms", *Parallel Problem Solving from Nature 2*, 37-46, 1992b.
- Gordon, W., "Sinectics: The development of the creative capacity", NY: Harper & Row, 1961.
- Gordon, T., Bentley, P., "On Evolvable Hardware", *Soft Computing in Industrial Electronics*, Physica-Verlag, 2002.
- Graf, J., Banzhaf, W., "Interactive evolution of images", *Proceedings of the Fourth Annual Conference on Evolutionary Programming*, pp. 53-65, 1995.
- Grefenstette, J. J., "Strategy Acquisition with genetic Algorithms", *The Handbook of Genetic Algorithms*, Ch. 12, Van Nostrand Reinhold, New York, pp. 186-201, 1991.
- Gugelot, H., "Praxis des Industrial Design", Ulm, n. 7, pp. 3-5, 1963.
- Guillot, A., Meyer, J.A., "Computer Simulations of Adaptive Behavior in Animats", *Proc. Computer Animation '94*, IEEE Computer Society Press, 1994.
- Gropius, W., Wensinger A., "The Theater of the Bauhaus", Baltimore: Johns Hopkins University Press, 1961.
- Hameyer, K. & Belmans, R., "Stochastic Optimisation of Mathematical Models for Electric and Magnetic Fields", *Proceedings of Adaptive Computing in Engineering Design and Control - '94*, pp. 139-144, 1996
- Harvey, I., Husbands, P., Cliff, D., "Issues in Evolutionary Robotics", From Animals to Animats 2, *Proceedings of the Second International Conference on Simulation of Adaptive Behavior (SAB92)*, MIT Press, Cambridge, MA, pp. 364-373, 1993.
- Harvey, I., Husbands, P., Cliff, D., "Seeing the light: artificial evolution, real vision", *Proceedings of the third international conference on Simulation of adaptive behavior : from animals to animats 3*, pp. 392-401, Brighton, 1994.
- Harvey, I., "Cognition is not Computation: Evolution is not Optimisation", *Proc. of 7th International Conference on Artificial Neural Networks ICANN97*, Springer-Verlag LNCS 1327, pp. 685-690, 1997.

Harvey, I., Thompson, A., "Through the Labyrinth Evolution Finds a Way: A Silicon Ridge", *Proceedings of the 1st Int. Conf. on Evolvable Systems: from Biology to Hardware (ICES96)*, Springer Verlag LNCS 1259, pp. 406-422, 1997.

Healey, A.J., Marco, D.B., "Slow speed flight control of autonomous underwater vehicles, experimental results with NPS AUV II", *Proceedings ISOPE*, pp. 523-532, 1992.

Healey, A.J., "A Neural Network Approach to Failure Diagnostics for Underwater Vehicles", *IEEE AUV'92*, pp. 131-134, 1992.

Healey, A.J., Lienard, D., "Multivariable Sliding Mode Control for Autonomous Diving and Steering of Unmanned Underwater Vehicles", *IEEE J. of Oceanic Engr.*, Vol. 18, No. 3, pp. 327-339, 1993.

Healey, A.J., Rock, S.M., Cody, S., Miles, D., Brown, J.P., "Toward an Improved Understanding of Thruster Dynamics for Underwater Vehicles", *Proc. AUV Symp.*, Boston, MA, pp. 340-352, 1994.

Herdy M., "Evolutionary Optimisation based on Subjective Selection – evolving blends of coffee", *Proceedings 5th European Congress on Intelligent Techniques and Soft Computing (EUFIT'97)*, pp. 640-644, 1997.

Higuchi, T., Niwa, T., Tanaka, T., Iba, H., de Garis, H., Furuya, T., "Evolving hardware with genetic learning", *Proc. of Simulation of Adaptive Behaviour*, pp. 417-424, MIT Press, 1992.

Higuchi, T., Iwata, M., Liu, W., "Machine Learning Approach to Gate-Level Evolvable Hardware", Tsukuba, Japan, Springer-Verlag, 1996.

Hine, B., Stoker, C., Sims, M., Rasmussen, D., Hontalas, P., "The Application of Telepresence and Virtual Reality to Subsea Exploration", *International Advanced Robotics Programme (IARP): Mobile Robots for Subsea Environments*, Monterey Bay Aquarium Research Institute, 1994.

Iba, H., Iwata, M., Higuchi, T., "Gate-Level Evolvable Hardware: Empirical Study and Application", *Evolutionary Algorithms in Engineering Applications*, pp.259-276, Springer Verlag, 1997.

Holland, J.H., "Adaptation in Natural and Artificial Systems", University of Michigan Press, 1975.

Holland, J.H., "The Royal Road for Genetic Algorithms: Fitness Landscapes and GA Performance", *Proceedings of the First European Conference on Artificial Life*, Cambridge, MA: MIT Press, pp.1-3, 6-7, 1991.

Holland, J.H., "Genetic Algorithms", *Scientific American*, pp. 66-72, 1992.

Holland, J.H., "Hidden order: how adaptation builds complexity", Addison Wesley, Reading Massachussets, 1995.

Horn, J., "Finite Markov Chain Analysis of Genetic Algorithms with Niching", *Proceedings of the Fifth International Conference on Genetic Algorithms*, Morgan Kaufmann Pub., pp. 110-17, 1993a.

- Horn, J., Nafpliotis, N., "Multiobjective Optimisation Using the Niche Pareto Genetic Algorithm", *Illinois Genetic Algorithms Laboratory (IlligAL)*, report no. 93005, 1993b.
- Horn, J., "Implicit Niching in a Learning Classifier System: Nature's Way", *Illinois Genetic Algorithms Laboratory (IlligAL)*, report no. 94001, 1994.
- Horner, A., Goldberg, D.E., "Genetic Algorithms and Computer-Assisted Music Composition", *Fourth International Conference on Genetic Algorithms*, pp.437- 441, 1991.
- Hover, F.S., Grosenbaugh, M.A., Triantafyllou, M.S., "Calculation of Dynamic Motions and Tensions in Towed Underwater Cables", *IEEE Journal of Oceanic Engineering*, vol. 19 no. 3, pp. 449-457, 1994.
- Huang, T.T., Feldman, J.P., "Development of Mathematical Models by Experimental and Analytical Methods for Predicting the Stability", *Control and Maneuvering Characteristics of Submarines*, technical report, Bethesda Maryland, 1988.
- Husbands, P., Harvey, I., Cliff, D., Thompson, A., Jakobi, N., "The Artificial Evolution of Robot Control Systems", *Proc. of the 2nd Int. Conf. on Adaptive Computing in Engineering Design and Control 96*. University of Plymouth, UK, pp. 41-49, 1996a.
- Husbands, P., Jermy, G., McIlhagga, M., Ives, R., "Two Applications of Genetic Algorithms to Component Design", *Selected Papers from AISB Workshop on Evolutionary Computing*, Springer-Verlag, Lecture Notes in Computer Science, pp. 50-61, 1996b.
- Ishii, K., Fujii, T., Ura, T., "Neural Network System for On-line Controller Adaptation and Its Application to Underwater Robot", *Proc. ICRA*, pp. 756-761, 1998.
- Islas C.A., Coello, E., Hernández, A., "Extraction of design patterns from evolutionary algorithms using case-based reasoning", *Evolvable Systems: From Biology to Hardware (ICES'2001)*, volume 2210, Springer-Verlag, Lecture Notes in Computer Science, pp 244-255, Tokyo, 2001.
- Jackson, H.A., "Fundamentals of Submarine Concept Design", *Society of Naval Architects and Marine Engineers (SNAME) Annual Conference*, pp. 28-31, 15-22, New York, 1992.
- Jang, D., Whigham, P.A., Dick G., "On evolving fixed pattern strategies for Iterated Prisoner's Dilemma", *Proceedings of the 27th conference on Australasian computer science*, Volume 26, New Zealand, pp: 241 - 247 , 2004.
- Jones J.C, Thornely, D., "Conference on Design Methods", Oxford, Pergamon Press, 1963.
- Jones, J.C., "Design Methods," London: Wiley-Interscience, 1970.
- Jones, J.C, "Métodos de diseño", Barcelona, Ed. Gustavo Gili, 1985.
- Kalganova, T., Miller J., "Evolving more efficient digital circuits by allowing circuit layout evolution and multi-objective fitness", *Proc. of The First NASA/DoD Workshop on Evolvable Hardware (EH'1999)*, IEEE Computer Society, California, 1999.

Kalganova T., "An extrinsic function-level evolvable hardware approach", *Proc. of the Third European Conference on Genetic Programming*, EuroGP2000, Springer-Verlag, Edinburgh, UK., 2000a.

Kalganova, T., "Bidirectional incremental evolution in evolvable hardware", *Proc. of The Second NASA/DoD Workshop on Evolvable Hardware (EH'2000)*, IEEE Computer Society, Palo Alto, California, 2000b.

Kanal, L., Cumar, V., "Search in Artificial Intelligence", Springer-Verlag Pub, 1988.

Kargupta, H. & Deb, K. & Goldberg, D., "Ordering Genetic Algorithms and Deception. Parallel Problem Solving from Nature", 22, Elsevier Science Publishers, pp. 47-56, 1992.

Kargupta, H., "Information Transmission in Genetic Algorithm and Shannon's Second Theorem", *Proceedings of the Fifth International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, 1993.

Karnaugh, M., "A map method for synthesis of combinational logic circuits", *Transactions of the AIEE, Communications and Electronics*, 72(1), pp. 593-599, 1953.

Kato S., "An image retrieval method based on a genetic algorithm controlled by user's mind", *Journal of the CRL*, Vol.48, No.2, pp.71-86, 2001.

Keane, A., "Experiences with optimizers in structural design", *Proceedings of the Conference on Adaptive Computing in Engineering Design and Control '94*, ed. I. C. Parmee, Plymouth, pp. 14-27, 1994.

Keane, A., Brown, S.M., "The design of a satellite boom with enhanced vibration performance using genetic algorithm techniques", *Proceedings of the Conference on Adaptive Computing in Engineering Design and Control 96*, ed. I. C. Parmee, P.E.D.C., ISBN 0 905227 61 1, pp. 107-113, Plymouth, 1996.

Kim, H.S., Cho, S.B., "Application of interactive genetic algorithm to fashion design", *Engineering Applications of Artificial Intelligence* 13(6), pp. 635-644, 2000.

Knjazew, D., Goldberg D.E., "Solving permutation problems with the ordering messy genetic algorithm", *Advances in evolutionary computing: theory and applications*, Springer-Verlag New York, 2003.

Kodogiannis, V.S., "Neural Network Adaptive Control for Underwater Robotic Systems", Mechatronics Group, Dep of Computer Science, University of Westminster. London, 2001.

Koza, J.R., "Genetic Programming: A paradigm for genetically breeding populations of computer programs to solve problems", *Technical Report STAN-CS-90-1314*, Stanford University, 1990.

Koza, J.R., "Genetic Programming: On the Programming of Computers by Means of Natural Selection", The MIT Press, Cambridge, 1992.

Krasnogor N., Gustafson S., "The Local Searcher as a Supplier of Building Blocks in Self-Generating Memetic Algorithms", *Fourth International Workshop on Memetic*

Algorithms (WOMA IV), Workshop Proceedings of the 2003 Genetic and Evolutionary Computation Conference, GECCO 2003, Chicago, 2003.

Kuroda, Y., Ura, T., Aramaki, K., "Vehicle Control Architecture for Operating Multiple Vehicles", *Proceedings of the IEEE Oceanic Engineering Society Conference Autonomous Underwater Vehicles (AUV) 94*, Cambridge Massachusetts, pp. 323-329, 1994.

Laurel, B., "The art of human computer interface design", Addison Wesley, EE.UU., May, 1990.

Layzell P., Thompson A., "Understanding Inherent Qualities of Evolved Circuits: Evolutionary history as a predictor of fault tolerance", *Proc. 3rd Int. Conf. on Evolvable Systems (ICES 2000): From biology to hardware*, Springer-Verlag, LNCS 1801 Ed. J. Miller and A. Thompson and P. Thomson and T. Fogarty, pp 133-144, 2000.

Levine, D., "Parallel genetic algorithm for the set partitioning problem", *Meta-heuristics: theory and applications*, Dordrecht: Kluwer Academic Publishers, 1996.

Levine, D., "A genetic algorithm for the set partitioning problem", *Proceedings of the 5th International Conference on Genetic Algorithms*, pp. 481-487, 1993.

Lewis, E.V., "Principles of Naval Architecture, volume III", *Society of Naval Architects and Marine Engineers (SNAME)*, New Jersey, pp. 188-190, pp. 418-423, 1988,

Lewis, F.L., Liu, K., Yesildirek, A., "Neural net robot controller with guaranteed tracking performance", *IEEE Trans. Neural Networks*, 6, pp. 703 -715, 1995.

Lohn, J., Reggia, J., "Discovery of Self-Replicating Structures Using a Genetic Algorithm", *1995 IEEE Int. Conf. on Evolutionary Computation (ICEC '95)*, vol 1, Perth, Western Australia, pp. 678-683, 1995.

Lomborg, B., "The Structure of Solutions in the Iterated Prisoner's Dilemma", Institute of Political Science, University of Copenhagen, 1991.

Lorentz, J., Yuh, J., « A Survey and Experimental Study of Neural Network AUV Control", *IEEE AUV96*, Monterey, 1996.

Louis, S.J., Rawlins G., "Designer genetic algorithms: Genetic algorithms in structure design", *Proc. of the Fourth International Conference on Genetic Algorithms*, pp 53-60, San Diego, 1991.

Louis, S.J., "Genetic Algorithm as a Computational Tool for Design", PhD thesis, Departament of Computer Science, Indiana University, August 1993.

Lund, H., Pagliarini, L., Miglino, O., "Artistic Design with GA and NN", *Proc. of the 1st Nordic Workshop on Genetic Algorithms and Their Applications (1NWGA)*, Uni. Vaasa, Finland, xiii+417 pp. 97-105, 1995.

Machado, P., Cardoso, A., "All the truth about NEvAr", *Applied Intelligence, Special issue on Creative Systems*, Vol. 16, Nr. 2, pp. 101-119, , 2002.

Maher, M.L., Zhao, F., Gero, J. S., "Creativity in humans and computers", 1989.

- Mahesh, M., Yuh, J., Lakshmi, R., "A coordinated control of an underwater vehicle and robotic manipulator", *J. of Robotic Systems on Underwater Robotics*, Vol. 8, No. 3, pp. 339-370, 1991.
- Mahfoud, S.W., "Finite Markov Chain Models of an Alternative Selection Strategy for the Genetic Algorithm", *Complex Systems* 7, pp. 155-70, 1993.
- Mahfoud, S.W., "Simple Analytical Models of genetic Algorithms for Multimodal Function Optimisation.", *Fifth international Conf. on Genetic Algorithms*, pp. 643, 1993.
- Maldonado, T., "El diseño industrial reconsiderado", Barcelona, Editorial Gustavo Gili, 1993, 1ª ed. 1977.
- Mañá, J., "El diseño Industrial. Barcelona", Salvat Editores, S.A., Tomo 2, pp. 108-109, 1973.
- McCluskey, E.J., "Minimization of boolean functions, *Bell Systems Technical Journal*, 35(6)", pp. 1417-1444, 1956.
- McCormack, J., "Interactive Evolution of L-System Grammars for Computer Graphics Modelling", *Complex Systems: from Biology to Computation*, chapter 21, pp. 118-130, IOSPress, Amsterdam, 1993.
- McLain, T.W., Rock, S.M., Lee, M.J., "Experiments in the coordinated control of an underwater arm/vehicle system", *Underwater Robots* (Eds. Yuh, Ura and Bekey), Kluwer, Boston, pp.139-158, 1996.
- McMillan, S., Orin, D.E., McGhee, R.B., "DynaMechs: An object oriented software package for efficient dynamic simulation of URVs", *Underwater Robotic Vehicles: Design and Control*, Albuquerque: TSI, 1995.
- Michalewicz, Z., "Genetic Algorithms + Data Structures = Evolution Programs", New York: Springer-Verlag, 1992.
- Michielssen, E., Ranjithan S., Mittra R., "Optimal Multilayer Filter Design Using Real Coded Genetic Algorithms," *IEEE Proceedings Part J. - Optoelectronic*, vol. 139, pp. 413-420, 1992.
- Miller, J.F., Thomson P., Fogarty T., "Designing electronic circuits using evolutionary algorithms. arithmetic circuits: A case study", *Genetic Algorithms and Evolution Strategy in Engineering and Computer Science*, pp. 105-131, Morgan Kaufmann, Chichester, England, 1997.
- Miller J., Kalganova, T., Fogarty, T., "Some aspects of an evolvable hardware approach for multiple-valued combinational circuit design", *Proc. of Second International Conference on Evolvable System: From Biology to Hardware (ICES'98)*, pp. 78-89, Lausanne, 1998.
- Miller, J., Kalganova, T., Lipnitskaya, N., "Multiple-valued combinational circuits synthesised using evolvable hardware approach", *Proc. of the 7th Workshop on Post-Binary Ultra Large Scale Integration Systems (ULSI'98)* in association with ISMVL'98, Fukuoka, Japan, 1998.

- Miller, J.F., Job, D., Vassilev, V.K., "Principles in the evolutionary design of digital circuits-part I", *Genetic Programing and Evolvable Machines*, 1(1/2):pp. 7-35, 2000.
- Mills, D., Harris, C.J., "Neurofuzzy modelling and control of a six degree of freedom AUV", Prentice Hall, Helmsel Hamstead, 1994.
- Minsky, M. L., "The Society of Mind", MIT Press, 1985.
- Monmarche, N., Nocent, G., Slimane, M., Venturini, G., Santini P., "Imagine: a tool for generating HTML style sheets with an interactive genetic algorithm based on genes frequencies," *Proc. 1999 IEEE Int. Conf. on Systems, Man and Cybernetics (SMC'99)*, pp 640-645, IEEE Press, NJ, 1999.
- Moore, J.H., "GAMusic: Genetic algorithm to evolve musical melodies", <http://www-cgi.cs.cmu.edu/afs/cs/project/airepository/ai/areas/genetic/ga/systems/gamusic/0.html>, 1994.
- Moore, M., "An Accurate and Efficient Parallel Genetic Algorithm to Schedule Tasks on a Cluster", *Proceedings of the 17th International Symposium on Parallel and Distributed Processing*, Page: 145, ISBN:0-7695-1926-1, IEEE Computer Society Washington, 2003.
- Moshfreti-Torbati, M., Keane, A., Brennan S., Rogers, E., "The integration of advanced active and passive structural vibration control", *Proceedings of VETOMAC-I*, Bangalore, India, October, 2000.
- Mount, J., "John Mount's International Interactive Genetic Art II", <http://www.geneticart.org/cgi-bin/mjwgenformII>, 1998.
- Mühlenbein, H., "How genetic algorithms really work I. Mutation and hillclimbing", *Parallel Problem Solving from Nature II*, pp. 15-25, Amsterdam, 1992.
- Munari, B., "¿Cómo nacen los objetos?", Barcelona. Ed. Gustavo Gili, p. 19, 1983.
- Murakawa M., Yoshizawa S., Kajitani I., Furuya T., Iwata M., Higuchi T., Hardwax Evolution at Function Level, *Proc. of the 4th Conf. on Parallel Problem Solving from Nature*, Berlin, , pp. 62-71, 1996.
- Musker, A.J., Loader, P.R., Butcher, M.C., "Simulation of a Submarine Under Waves," *International Shipbuilding Programme*, vol. 35 no. 404, pp. 389-410, 1988.
- Nakamura, Y., Savant, S., "Nonlinear Tracking Control of Autonomous Underwater Vehicles", *Proceedings of IEEE Int. Conf. on Robotics and Automation*, Vol. 3, pp. A4-A9, 1992.
- Nash, W. A., *Hydrostatically Loaded Structures*. Pergamon, Oxford, 1995.
- Negroponte N., "Being Digital". *Vintage books*, Nueva York, NY, 1994
- Ngo, J.T., Marks J., "Spacetime Constraints Revisited. Computer Graphics", *Annual Conference Series*, pp. 335-342, 1993.
- Nielsen, J., "Usability engineering". *AP Professional*, Boston, MA, 1993.

- Norman, D., "The design of everyday things", Currency/Doubleday, NY, 1988.
- Okay, M., Takagi, H., "Application of Interactive Evolutionary Computation to Optimal Tuning of Digital Hearing Aids," *5th Int'l Conf. on Soft Computing (IIZUKA'98)*, pp.849-852, World Scientific, Japan, pp. 16-20, 1998.
- Orrick, A., McDermott, M., "Failure Detection in an Autonomous Vehicle", *Proceedings of IEEE Symposium on Autonomous Underwater Vehicle Technology*, AUV '94, pp. 377-382, 1994.
- Osborn, A.F., "Your creative potencial", Charles Scribner's Sons, New Cork, 1948.
- Oyama, A., Obayashi, S., Nakamura, T., "Real-Coded Adaptive Range Genetic Algorithm Applied to Transonic Wing Optimization", *Parallel Problem Solving from Nature 6th International Conference*, Springer Verlag, Paris, France, 2000.
- Pappas, G., Shotts, W., O'Brien, M., Wyman, W., "The DARPA/Navy Unmanned Undersea Vehicle Program," *Unmanned Systems*, vol. 9 no. 2, pp. 24-30, 1991.
- Park, B., Messer, C.J., Urbanik T., "Enhanced Genetic Algorithm for Signal Timing Optimization of Oversaturated Intersections", *Transportation Research Record 1727*, TRB, National Research Council, Washington, pp. 32-41, 2000.
- Parmee, I. C., Denham M. J., "The Integration of Adaptive Search Techniques with Current Design Practice", *Proceedings of Adaptive Computing in Engineering Design and Control*, University of Plymouth, UK, pp. 1,13, 1994.
- Parmee, I.C., "Evolutionary and adaptive computing in engineering design", Springer-Verlag, London, 2001.
- Parmee, I.C., "The Concrete arch dam: An evolutionary model of the design process," *Int. Conf on Artificial neural Nets and Genetic Algorithms*, Austria, pp. 544551, 1993.
- Parmee, I., "The Development of a Dual-Agent Strategy For Efficient Search Across Whole System Engineering Design Hierarchies", *4th Int. Conf. on Parallel Problem Solving From Nature*, Berlin, pp. 22-27, 1996.
- Parmee, I., "Towards an optimal Engineering Design Process using Appropriate Adaptive Search Strategies", *Journal of Engineering Design*, Vol 7:4, Carfax Pub,1996.
- Park, B., Messer, C. J., Urbanik II, T., "Enhanced Genetic Algorithm for Signal Timing Optimization of Oversaturated Intersections", *Transportation Research Record 1727*, TRB, National Reserach Council, Washington DC, pp. 32-41, 2000
- Patch, D. A., "Solar Energy System for Long-Term Deployment of AUVs", *International Unmanned Undersea Vehicle Symposium*, 2000.
- Paton, R., "Enhancing Evolutionary Computation using Analogues of Biological Mechanisms", *Evolutionary Computing*, AISB Workshop, pp. 51-64, Springer-Verlag, Germany, 1994.
- Pham, D. T., Yang, Y., "A genetic algorithm based preliminary design system", *Journal of Automobile Engineers* v207:D2, pp. 127-133, 1993.

- Poli R., Langdon, W.B., "Genetic Programming with One-Point Crossover and Point Mutation", 1997.
- Poli, R., Langdon, W.B., "Foundations of Genetic Programming", Springer, 2002.
- Quarante, D., "Diseño Industrial", Barcelona, Ediciones CEAC, S.A, Tomo 1., pp. 59-75. Tomo II, 1992.
- Quine W. V., "A way to simplify truth functions", *American Mathematical Monthly*, 62(9), pp.627-631, 1955.
- Radcliffe, N. J., Surry, P. D., "Co-operation through Hierarchical Competition in Genetic Data Mining", *Parallel Problem Solving From Nature*, 1994.
- Rechenberg, I., "Cybernetic Solution Path of an Experimental Problem", Royal Aircraft Establishment, Library Translation No. 1122, Farnborough, England, 1964.
- Rechenberg, I., "Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution", Stuttgart: Fromman-Holzboog Verlag, 1973.
- Ricard, A., "Diseño y calidad de vida", Barcelona, Fundación BCD, 1985.
- Poli, R., Page, J., "Solving high-order boolean parity problems with smooth uniform crossover, sub-machine code gp and demes", *Genetic Programming and Evolvable Machines*, 1(1-2), pp.37-56, 2000.
- Rodríguez K., Fleming, P. J., "Functionality and Optimality in Circuit Design: A Genetic Programming Approach", *Proceedings of the Third International Symposium on Adaptive Systems, Evolutionary Computation and Probabilistic Graphical Models*, pp. 23-28, Cuba, 2001.
- Rosenbluth, A., Wiener, N., "Behaviour, Purpose and Teleology", *Philosophy of Science*, Vol. 10, pp 18-24, 1945.
- Rosenman, M. A., Gero, J. S., "Modelling multiple views of design objects in a collaborative CAD environment", *CAD, Special Issue on AI in Design* 28(3), pp. 207-21, 1996.
- Rosenman, M., "The Generation of form Using an Evolutionary Approach", *Evolutionary Algorithms in Engineering Applications*, Springer-Verlag Pub., pp. 69-86, 1997.
- Roston, G., "Hazards in Genetic Design Methodologies", *Evolutionary Algorithms in Engineering Applications*, Springer-Verlag Pub., pp. 135-154, 1997.
- Rowland, D., Biocca, F., "Evolutionary co-operative design between human and computer: implementation of the genetic sculpture park", *Proceedings of the fifth symposium on Virtual reality modeling language (Web3D-VRML)*, 2000.
- Salmon, R., Slater, M., "Computer Graphics: Systems and Concepts". Addison-Wesley, 1987.

- Santos, A., Dorado, J., Romero, J., Arcay, B., Rodríguez, J., "Artistic Evolutionary Computer Systems", *Proceedings of the Genetic and Evolutionary Computation Conference Workshop*, Las Vegas, 2000.
- Sato, Y., "Voice Conversion Using Interactive Evolution of Prosodic Control", *Proc. of the 2002 Genetic and Evolutionary Computation Conference*, pp. 1204-1211, New York, 2002.
- Savic, D. A. e Walters, G. A.; "Genetic algorithms for least-cost of water distribution networks", *Journal of Water Resources Planning and Management*, ASCE, Vol. 123 N° 2, New York, pp 67-77, 1997.
- Schaffer, J.D., "Multiple Objective Optimization with Vector Evaluated Genetic Algorithms", PhD thesis, Vanderbilt University, 1984.
- Schaffer, J.D., Eshelman, L., "Combinatorial Optimization by Genetic Algorithms: The Value of Genotype/Phenotype Distinction", *Proc. of Applied Decision Technologies (ADT '95)*, London, pp. 29-40, 1995.
- Schaffer, J.D., "Multiple Objective Optimization with Vector Evaluated Genetic Algorithms", *Genetic Algorithms and Their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pp. 93-100, 1985.
- Schnier, T., Gero, J.S., "Learning genetic representations as an alternative to hand-coded shape grammars", *Artificial Intelligence in Design '96*, Kluwer, Dordrecht, pp. 39-57, 1996.
- Schoenauer, M., "Shape Representations and Evolution Schemes", *Proc. of the 5th Annual Conf. on Evolutionary Programming*, MIT Press, Cambridge, pp.121-129, 1996.
- Schwefel H.P., "Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie", Basel: Birkhäuser, 1977.
- Schwefel H.P., "Evolutionsstrategie und numerische Optimierung", PhD thesis, Technical University of Berlin, Department of Process Engineering, Berlin, 1975.
- Shannon, R.E., "Systems Simulation: the art and science", Prentice-Hall, Inc, 1975.
- Shiraki H., Saito H., "An Interactive Image Retrieval System using Genetic Algorithms." *Procs of International Conference on Virtual Systems and Multimedia (VSMM'96)*, pp 257-262, 1996.
- Shoults, G.A., "Dynamics and Control of an Underwater Robotic Vehicle with an N-axis Manipulator", Ph.D. Thesis, Washington University, 1996.
- Sims, K., "Evolving 3D Morphology and Behaviour Schemes", *Proceedings of the 5th Annual Conference on Evolutionary Programming*, Cambridge, MA: MIT Press, pp. 121-129, 1994
- Sims, K., "Evolving Virtual Creatures. In Computer Graphics", *Annual Conference Series (SIGGRAPH '94 Proceedings)*, pp. 15-22, 1994.
- Sims, K., "Evolving three-dimensional Morphology and Behaviour", *Evolutionary Design by Computers*, Ch. 13, 1999.

- Sims, K., "Artificial Evolution for Computer Graphics", *Computer Graphics*, 25, No.4, 319-328, 1991.
- Sims, K., "Evolving 3D Morphology and Behaviour by Competition", *Artificial Life IV Proceedings*, MIT Press, pp. 28-39, 1994.
- Sims, K., "Evolving Virtual Creatures", *Computer Graphics, Annual Conference Series, (SIGGRAPH '94 Proceedings)*, pp. 15-22, 1994.
- Smallwood, D.A., Whitcomb, L.L., "Toward Model Based Trajectory Tracking of Underwater Robotic Vehicles: Theory and Simulation", 21th International Symposium on Unmanned Untethered Submersible Technology, 2001.
- Smith, R.E., Goldberg, D.E., "Diploidy and Dominance in Artificial Genetic Search", *Complex Systems* 6, pp. 251-285, 1992.
- Spears, W.M., De Jong, K.A., Bäck, T., Fogel, D.B., de Garis, H., "An Overview of Evolutionary Computation", *Proceedings of the European Conference on Machine Learning (ECML-93)*, 1993.
- Srinivas, N., Deb, K., "Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms", *Evolutionary Computation*, v2:3, pp. 221-248, 1995.
- Stephens, C.R., Waelbroeck, H., "Effective degrees of freedom in genetic algorithms and the block hypothesis", *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA97)*, pp. 34-40, 1997.
- Stephens, C.R., Waelbroeck, H., "Schemata evolution and building blocks", *Evolutionary Computation*, 7(2):109-124, 1999.
- Suto T., Ura, T., "Unsupervised Learning System for Vehicle Guidance Constructed with Neural Network", *Proc. 8th International Symposium on Unmanned Untethered Submersible Technology*, pp. 222-230, 1993.
- Tabaai, S.S., El-Hawary, F., El-Hawary, M., "Hybrid Adaptive Control of Autonomous Underwater Vehicle." *Proc. AUV Symp*, pp. 275-282, 1994.
- Taboada, E., Nápoli, R., "El diseño industrial", Buenos Aires, Centro Editor de América Latina, S.A., pp. 59, 1977.
- Takagi H, Ohsaki M., "Automatic optimization of hearing aids based on how user hears," *Annual Meeting of Acoustic Society of Japan*, 1-2-18, pp.359-360, 1999.
- Takagi, H., "Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation", *Proceedings of the IEEE*, 89(9), pp.1275-1296, 2001.
- Takai, M., Ura, T., "A Model Based Self Diagnosis System for Autonomous Underwater Vehicles", *IIS, University of Tokyo*, 1994.
- Tennant, A., Chambers, B., "Adaptive Optimisation Techniques for the Design of Microwave Absorbers", *Proceedings of Adaptive Computing in Engineering Design and Control '94*, pp. 44-49, 1994.

- Thompson, A., "Evolving Fault Tolerant Systems. In Genetic Algorithms in Engineering Systems: Innovations and Applications", *IEE Conf. Pub. No. 414*, pp. 524-529, 1995.
- Thompson, A., "Notes on design through artificial evolution: Opportunities and algorithms", *Adaptive computing in design and manufacture*, 2000.
- Thomson P., Miller J.F., "Evolving digital electronic circuits for real-valued function generation using a genetic algorithm", *Genetic Programming 1998: Proceedings of the Third Annual Conference*, pp 863–868, 1998.
- Todd, S., Latham, W., *Evolutionary Art and Computers*, Academic Press, 1992.
- Torresen J., "Evolvable hardware as a new computer architecture. ," Proc. of the International Conference on Advances in Infrastructure for e-Business, e-Education, e-Science, and e-Medicine on the Internet, 2002.
- Tsukamoto, C.L., Yuh, J., Choi, S.K., Lee, W.C., Lorentz, J., "Experimental Study of Advanced Controllers for an Underwater Robotic Vehicle Thruster System2, International J. of Intelligent Automation and Soft Computing, Vol. 5, No. 3, pp. 225-238, 1999.
- Tudela, F., "Introducción" y "Arquitectura e ideología" en *Arquitectura y procesos de significación*, Edicol, México, 1980.
- Tuohy, S.T., "A Simulation Model for AUV Navigation", *Proceedings of the IEEE Oceanic Engineering Society Conference Autonomous Underwater Vehicles (AUV) 94*, pp. 470-478, 1994.
- Ulrich, K., Seering W., "A Computational Approach to Conceptual Design", *Proceedings, International Conference on Engineering Design*, 1987.
- Unemi, T., "SBART 2.4: an IEC Tool for Creating 2D images, movies and collage", *Proceedings of the Genetic and Evolutionary Computation Conference Program*, Las Vegas, 2000.
- Valavanix, K.P., Gracanin, D., Matijasevic, M., Kolluru, R., Demetriou G.A., "Control Architectures for Autonomous Underwater Vehicles", *IEEE Control Systems Magazine*, pp. 48-64, 1997.
- Van Doesburg, T., "Towards a Collective Construction", *De Stijl*, 1923.
- Van Veldhuizen, D., Lamont, G., "Multiobjective optimization with messy genetic algorithms", *Proceedings of the 2000 ACM symposium on Applied computing*, Como, Italy, pp: 470 – 476, ISBN:1-58113-240-9, 2000.
- Ventrella, J., "Explorations in the Emergence of Morphology and Locomotion Behaviour in Animated Characters", *Artificial Live IV, Proc. of the 4th Int. Workshop on the Synthesis and Simulation of Living Systems*, MIT Press, pp. 436-441, 1994.
- Vico F.J., Veredas F.J., Bravo J.M., Almaraz J., "Automatic design synthesis with artificial intelligence techniques". *Artificial Intelligence in Engineering 13*, pp. 251-256, 1999.

Whitley, D., Starkweather, T., "GENITOR II: a distributed genetic algorithm", *Journal of Experimental and Theoretic Artificial Intelligence*, v2:3, pp. 189-214, 1990.

Williams, B.C., "Visualising potential interactions: constructing novel devices from first principles", *Proc. of Eighth National Conference on Artificial Intelligence (AAAI-90)*, 1990.

Wu, A. S., Garibay, I., "The proportional genetic algorithm: Gene expression in a genetic algorithm", *Journal Genetic Programming and Evolvable Hardware*, volume 3, pp. 157-192, 2002.

Yamada, T., Nakano, R., "A genetic algorithm with multi-step crossover for job-shop scheduling problems. In Genetic Algorithms in Engineering Systems: Innovations and Applications", *IEE Conf. Pub.* No. 414, pp. 146-151, 1995.

Yang, K.C., Yuh, J. and Choi, S.K., "Fault-Tolerant System Design of an Autonomous Underwater Vehicle -- ODIN: an Experimental Study", *International J. of Systems Science*, 1999.

Yoerger, D.N. and Slotine, J.E., "Robust Trajectory Control of Underwater Vehicles", *IEEE J. of Oceanic Engr.*, OE-10 (4), pp. 462-470, 1985.

Yoerger, D.N., Cooke, J.G., and Slotine, J.E., "The influence of thruster dynamics on underwater vehicle behavior and their incorporation into control system design", *IEEE J. Ocean Engineering*, OE-15 (3): pp. 167-178, 1990.

Yuh, J., "A neural net controller for underwater robotic vehicles", *IEEE J. Oceanic Engineering*, 15 (3), pp.161-166, 1990.

Yuh, J., "Learning control for underwater robotic vehicles", *IEEE Control System Magazine*, 14 (2), pp. 39-46, 1994.

Yuh, J., "A neural net controller for underwater robotic vehicles", *IEEE Journal of Oceanic Engineering*, 15(3), pp.161-166, 1990.

Zebulum, R., Stoica, A. and Keymeulen, D., "Mixtrinsic evolution", *Proc. of Third International Conference on Evolvable System: From Biology to Hardware (ICES 2000)*, pp 208-217, 2000.

GSA

Grupo de Sistemas Autónomos

En esta Tesis Doctoral se define un entorno modular que plantea el problema de diseño como una búsqueda automática de extremos en un espacio de soluciones variable de alta dimensionalidad. De este modo, basándose en la integración de las etapas del proceso de diseño, se reduce la participación del diseñador humano a la formulación y análisis del problema (especificación), liberándolo de su función clásica de búsqueda y toma de decisiones (diseño). Siguiendo esta nueva metodología se mejora la creatividad al eliminarse las restricciones ficticias introducidas por el diseñador, se hacen los problemas más manejables y se obtienen soluciones variadas de mayor calidad al trabajar directamente sobre los parámetros del dominio del problema en cuestión..

El entorno definido se basa en la aplicación de técnicas evolutivas de búsqueda y en simuladores avanzados de alta fiabilidad que se ejecutan mediante programación distribuida para aumentar la potencia de cálculo y dotar al sistema de escalabilidad. Estos simuladores evalúan técnicamente cada una de las soluciones o las visualizan para permitir la opinión subjetiva de expertos o usuarios mediante una interfaz hombre-máquina, confiriendo al entorno de cierto carácter concurrente.

Este cambio radical del uso de herramientas computacionales en el proceso de diseño transforma el concepto de herramientas de "ayuda al diseño" a un nuevo sistema de "ayuda al diseñador" que relega al humano a una función meramente especificadora y de control para la búsqueda y síntesis de posibles soluciones.